

Formation Planning of Mobile Robots in Changeable Assembly Systems

Formationsplanung mobiler Roboter in wandlungsfähigen Montagesystemen

Von der Fakultät für Maschinenwesen
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades einer
DOKTORIN DER INGENIEURWISSENSCHAFTEN
genehmigte Dissertation

vorgelegt von

Aline Chantal Kluge-Wilkes

Berichter:

Univ.-Prof. Dr.-Ing. Robert Schmitt
Prof. Ole Madsen, Ph.D.

Tag der mündlichen Prüfung: 22.11.2024

Acknowledgements

This thesis was written during my work as a research associate at the Chair of Intelligence in Quality Sensing (IQS) at the WZL of RWTH Aachen University as part of the research project EXC–2023 Internet of Production–390621612, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy.

First, I would like to thank my doctoral supervisor and examiner, Prof. Dr.-Ing. Robert Schmitt, for the opportunity to pursue my doctorate and for the trust placed in me. I express my gratitude to Prof. Ole Madsen, Ph.D., from the Department of Materials and Production of Aalborg University (AAU) for his keen interest in my work and for taking on the role of second examiner. Additionally, I extend my thanks to Prof. Dr.-Ing. Corves for presiding over the examination.

I extend my appreciation to my colleagues at the institute for their exceptional collaboration, open dialogue, and inspiring approach that has empowered us to shape our technological future actively. A particular acknowledgement goes to Dr.-Ing. Amon Göppert, thank you for your critical review and for being a supportive supervisor. I sincerely thank Dr.-Ing. Tobias Müller, for his open ear, long walks and who consistently provided constructive encouragement for my endeavours. My gratitude also goes to Dr.-Ing. Armin Buckhorst for years of fruitful collaboration in multiple projects and for setting a commendable example in project structuring. I want to extend my heartfelt gratitude to all of my colleagues of the "Diss-Selbsthilfegruppe" for their invaluable structural, content-related, and mental support throughout the conduction of research and the writing of the thesis. Moreover, I thank Lea Kaven, Jonas Baum and Dr.-Ing. Sören Münker for their feedback and corrections of the manuscript. Additionally, I want to express thanks to Guillaume Tréheux, Henrik Hose, and Lukas Schäper for their friendship. I want to thank Presley Demuner Reverdito, my student assistant, for his valuable contributions and year-long support in bringing this work to fruition.

Eventually, I would like to thank my family and friends who paved the way for this work and stood by me throughout the journey. My heartfelt thanks go to my parents, Bettina and Hans, and my siblings, Luise and Tim, for always supporting, helping and advising me. Finally, I would like to thank my fiancé Georgi for his open ear, ability to cheer me up, and understanding, especially towards the end of the dissertation. I am looking forward to build a family with you!

Aline Kluge-Wilkes

Abstract

The challenges of reduced product lifecycles in a global market, volatile supply chains, and labour shortages require increased adaptability and automation in assembly. The key obstacle to dynamic adaption and reconfiguration in assembly is the significant manual setup, programming, and control costs of mobile manipulators. This dissertation develops and validates a method for automatically evaluating and planning the positioning of mobile manipulators in assembly stations, ensuring the successful execution of all assembly tasks. Therefore, a formation planning methodology, comprising the three developed building blocks of description model for capability-based task allocation, executability evaluation and base placement is introduced.

Firstly, a syntactic and semantic description model of mobile manipulators and their capabilities is developed as a fundamental element in planning assembly stations. The "CAPaBILity-based resource AllocatiON Ontology" (CAPILANO) models assembly resources (mobile bases, robot arms, and end-effectors), their combinability, and the capability corridor of the combined resources. CAPILANO allows for the automatic deduction of a combination of resources that offer the required capabilities for task execution. The developed subsequent task allocation assigns the task to the best-suited combined resource.

Knowledge of the spatial distribution of performance metrics in the manipulator's workspace is fundamental to determining where mobile manipulators should be placed in relation to the product on which the assembly operation is to be carried out. Performance metrics quantify the executability of specific assembly tasks. The distribution of these metrics in the manipulator's workspace is visualised in the developed performance maps. The effect of equipped end-effectors on the executability is incorporated for the first time. Finally, capability-based task allocation and workspace evaluation are incorporated into the developed formation planning for mobile manipulators. Formations are heuristically determined, and the feasibility of the resulting formations is evaluated and weighted, resulting in a formation that ensures the executability of all tasks.

Formation planning is verified and validated by decomposition and submodel testing of the three artifacts of the description model, executability evaluation and robot base placement. The operational validity is empirically supported by simulations and demonstrations. The resulting findings confirm the executability of the planned formations and document a reduction in the time required for formation planning. Existing challenges lie in the temporal scheduling of task execution and computational performance for scenarios involving numerous products and tasks.

Kurzfassung

Reduzierte Produktlebensdauern, volatile Lieferketten und zunehmender Arbeitskräftemangel verstärken den Druck Produktion- und Montagelinien flexibler und automatisierter zu gestalten. Ein zentrales Hindernis der dynamischen Adaptierbarkeit und Rekonfiguration in der Montage ist der hohe manuelle Aufwand für die Einrichtung, Programmierung und Steuerung mobiler Manipulatoren. In dieser Dissertation wird eine Methode zur automatischen Bewertung und Planung der Positionierung von mobilen Manipulatoren in Montagestationen entwickelt und validiert, welche die Ausführbarkeit aller Montageaufgaben gewährleistet.

Zunächst wird ein syntaktisches und semantisches Beschreibungsmodell für mobile Manipulatoren und deren Fähigkeiten als grundlegendes Element für die Planung von Montagestationen konzeptioniert und entwickelt. Die "CAPaBLity-based resource AllocatioN Ontology" (CAPILANO) modelliert Montageressourcen (mobile Basen, Roboterarme und Endeffektoren), deren Kombinierbarkeit und den aufgespannten Fähigkeitskorridor der kombinierten Ressourcen. CAPILANO ermöglicht die automatische Deduktion der zusammengesetzten Ressourcen, welche in Kombination die für die Aufgabenerfüllung erforderlichen Fähigkeiten bereitstellen.

Die Kenntnis der räumlichen Verteilung von Performanzindikatoren im Arbeitsraum des Manipulators ist von grundlegender Bedeutung für die Bewertung der Ausführbarkeit und Platzierung mobiler Manipulatoren in Bezug auf das Produkt, an dem der Montagevorgang durchgeführt werden soll. Performanzindikatoren, die die Ausführbarkeit spezifischer Montageaufgabentypen quantifizieren, werden ermittelt. Für die Bestimmung der räumlichen Verteilung der Performanzindikatoren werden performance maps entwickelt. Hierbei wird die Auswirkung von ausgerüsteten Endeffektoren auf die Ausführbarkeit erstmalig einbezogen. Für die Formationsplanung wird eine Manipulatorplatzierung unter Einbeziehung der Aufgabenverteilung und Arbeitsraumbewertung entwickelt.

Die Formationsplanung wird durch die Zerlegung in drei Artefakte des Beschreibungsmodells, der Ausführbarkeitsbewertung und der Roboter-Basis-Platzierung und die Prüfung von diesen als Teilmodellen erfolgreich verifiziert und validiert. Mittels Simulationen und Demonstrationen wird die operationale Validität empirisch belegt. Die resultierenden Befunde bestätigen die Ausführbarkeit der geplanten Formationen und dokumentieren eine Verringerung der erforderlichen Zeit für die Formationsplanung. Bestehende Herausforderungen liegen in der zeitlichen Planung der Aufgabenausführung sowie in der Rechenleistung für Szenarien mit einer hohen Anzahl von Produkten und Aufgaben.

I Contents

I	Contents	i
II	List of figures	iv
III	List of tables	vi
IV	List of acronyms	vii
V	List of symbols	ix
1	Motivation, research methodology and structure	1
1.1	Need for formation planning of mobile robots	1
1.2	Research objective and application domain	3
1.3	Research methodology and derived structure	5
2	Knowledge base and foundations	10
2.1	Changeable assembly systems	10
2.1.1	History and terminology of changeable assembly	10
2.1.2	Mobile robots in changeable assembly	11
2.2	Workspace evaluation of mobile robots in assembly	14
2.2.1	Robot workspaces	14
2.2.2	Metrics to evaluate robot workspaces	17
2.2.3	Workspace representation with combined metrics	21
2.2.4	Robot base placement	23
2.3	Semantic description of assembly robot capabilities	24
2.3.1	Ontologies in the manufacturing domain	25
2.3.2	Capability models	27
2.3.3	Ontology development methodologies	29
2.4	Task clustering	30
2.5	Verification and validation methodology	33
3	Formation planning of mobile robots state of the art	35
3.1	Methodological approach to the systematic literature review	35
3.2	Qualitative literature study	37
3.3	Research deficits	42
3.4	Research scope and research question	43
4	Semantic description model of mobile robots and their capabilities	45
4.1	Applied methodology of building CAPILANO	46

4.1.1	Ontology development methodology	46
4.1.2	Application of the methodology	46
4.2	Need and feasibility phase	48
4.2.1	Requirements and competency questions	48
4.2.2	Consolidation of the state of the art	49
4.3	Ontology-based definition	52
4.3.1	Modelling of assembly resources	54
4.3.2	Modelling of robot capabilities	55
4.3.3	Implementing CAPILANO	57
4.4	Task allocation	58
4.5	Verification and validation of CAPILANO	62
4.6	Interim conclusion: Description model	65
5	Evaluation of the executability of assembly tasks by a robot	68
5.1	Evaluating a robot's workspace with different performance metrics	70
5.2	End-effector-dependent executability evaluation	76
5.2.1	Calculating the region of feasible robot flange poses	77
5.2.2	Average performance metric	80
5.3	Verification and validation of the workspace analysis	81
5.4	Interim conclusion: Executability evaluation	85
6	Autonomous clustering-based task allocation and robot base placement	87
6.1	Requirements and preconditions	88
6.2	Motivating and choosing clustering algorithms	89
6.3	Autonomous task clustering and base placement	90
6.3.1	Task types	92
6.3.2	Search space	92
6.3.3	Task clustering	94
6.3.4	Favoured base placements	96
6.3.5	Selection of base placements	97
6.4	Verification and validation of the base placement and task clustering	101
6.5	Interim conclusion: Clustering-based task allocation and base placement	107
7	Verification and validation	111
7.1	Conceptual model validation	113
7.2	Computerised model verification	114
7.3	Operational validation	116
7.4	Data validity	119
7.5	Interim conclusion: Verification and validation	119

8	Conclusion and critical reflection	121
8.1	Reflection on the findings	121
8.2	Reflection on the research methodology	123
9	Summary and outlook	125
9.1	Summary	125
9.2	Future research and outlook	129
	References	133
	List of related publications	158
	List of contributing theses	161
Appendix A	Motivation, research methodology and structure	162
Appendix B	Methodological approach to the systematic literature review . .	163
Appendix C	Ontology-based definition	165
Appendix D	Workspace evaluation of mobile robots in assembly	167
Appendix E	Evaluation of the executability of assembly tasks by a robot . .	175
Appendix F	Autonomous clustering-based task allocation and robot base placement	180
Appendix G	Verification and validation	185

II List of figures

1.1	Heuristic reference frame relating formation planning to neighbouring domains.	4
1.2	Formation planning and its building blocks.	5
1.3	Structure of the thesis according to Design Science Research.	7
1.4	Research artifact development process.	8
1.5	Chronological thesis structure.	8
2.1	Hierarchic abstraction levels of operation planning.	12
2.2	Components of a mobile multipurpose robot.	13
2.3	Reachable workspace of a "decoupled" 6-axes robot.	15
2.4	Topology of a "decoupled" 6-axes robot.	16
2.5	Geometrical approach to workspace calculation.	17
2.6	Allowable cone of robot flange orientation.	19
2.7	Capability map of a UR10 robot arm.	22
2.8	Product, Process and Resource (PPR) model.	26
2.9	Modelling robot capabilities.	29
2.10	Ontology development as part of the digital twin pipeline.	30
2.11	Classification scheme for clustering algorithms.	31
3.1	Procedure of the systematic literature review.	36
3.2	Station formation problem and solution.	43
3.3	Mapping of building blocks to deficits to research questions.	44
4.1	Mapping of the phases of ontology development.	47
4.2	Reuse and influence of established ontologies on CAPILANO.	53
4.3	CAPaBLity-based resource AllocationN Ontology class model.	54
4.4	Modelling of semantic robot capabilities.	57
4.5	Conceptual model of robot capabilities.	58
4.6	Task allocation framework.	59
4.7	Ontology querying process.	61
4.8	Average total runtime of the task allocation.	64
4.9	Average runtime per query.	64
4.10	Contribution of chapter 4 to the research progress.	67
5.1	Process overview of evaluating the executability of assembly tasks by a robot.	69
5.2	Pose calculation for performance map creation.	73
5.3	Exemplary results of the generated performance map of a UR10.	74
5.4	Exemplary results of the generated performance map of an ABB IRB1600-1.2.	76
5.5	Process steps for evaluating the end-effector-dependent executability.	77
5.6	Determination of the feasible robot flange region.	78
5.7	Region of feasible robot flange poses.	79
5.8	Alternative gripping poses of a parallel gripper.	79
5.9	Average reachability indexes of a UR10.	82
5.10	Correlation of the average reachability index and the inverse kinematics.	82

5.11	Contribution of chapter 5 to the research progress.	86
6.1	Process steps to cluster task poses and determine robot base placements. . . .	91
6.2	CAD model of a turbine and the extracted mesh to create task poses.	92
6.3	Creation of a search space for possible base placements around the product. .	94
6.4	Search space discretisation around a turbine.	96
6.5	Extraction of favoured base placements.	97
6.6	Regions of feasible robot flange poses for a screwing task.	98
6.7	Deriving robot placements from task poses.	100
6.8	Results of different methods to create exemplary task poses on products. . . .	102
6.9	Time variance for base placement.	104
6.10	Assembly task poses on products.	105
6.11	Test scenarios with identical search space resolution and increasing task poses.	106
6.12	Operational validation: Increasing number of task poses for base placement. .	107
6.13	Contribution of chapter 6 to the research progress.	108
7.1	Simulation of the task execution.	116
7.2	Operational validation of base placement and task execution in a test set-up. .	118
9.1	Mapping the research results to the research deficits and research areas. . . .	126
9.2	Developed research results in the Internet of Production (IoP).	127
A.1	Design Science Research framework.	162
C.1	CAPaBILity-based resource AllocationN Ontology (CAPILANO) — class diagram.	166
E.1	Part 1: Results of the Design of Experiments of the Performance Maps.	177
E.2	Part 2: Results of the Design of Experiments of the Performance Maps.	178
E.3	Part 3: Results of the Design of Experiments of the Performance Maps.	179
F.1	Partially automated creation of distributed task poses.	180
F.2	Test scenarios with increasing search space resolution.	181
F.3	Test scenarios with constant search space resolution and increasing task poses.	182
F.4	Failed test scenario of base placement.	182
F.5	Part 1: Simulation of task execution.	183
F.6	Part 2: Simulation of task execution.	184
G.1	Metrological test setup.	185
G.2	Task poses indicated by stickers for marking on the turbine.	185

III List of tables

2.1	Excerpt of table D.1: Overview and classification of robot performance metrics.	21
3.1	Evaluation criteria for the literature review.	38
3.2	Results of the qualitative literature study.	41
4.1	Evaluation criteria of suitability of description models.	50
4.2	Suitability of ontologies and description models for formation planning.	50
4.3	Evaluation of CAPILANO.	65
5.1	Metrics for evaluation and metric categories.	70
5.2	Matching of performance metrics to tasks.	72
5.3	Design of experiment for conceptual model validation of the performance map.	84
6.1	Time variance of similar test scenarios of base placement.	104
6.2	Fulfilment level of requirements of task clustering and base placement.	109
7.1	Applied techniques of verification and validation.	112
7.2	Sensitivity analysis of the formation planning.	118
9.1	List of contributing theses.	161
B.1	Search strings and keywords for the systematic literature mapping and review.	163
B.2	Phases of the literature review.	164
D.1	Overview and classification of robot performance metrics.	167
E.1	Overview and classification of robot capabilities.	175

IV List of acronyms

AAS	Asset Administration Shell
AGV	Automated Guided Vehicle
BMS	Biological Manufacturing System
CAD	Computer-Aided Design
CAPILANO	CAPablLity-based resource AllocationN Ontology
CORA	Core Ontology for Robotics and Automation
C4I	Capability For Industry
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DIAS	Dynamically Interconnected Assembly System
DIN	Deutsches Institut für Normung e.V.
DMS	Dedicated Manufacturing System
DoF	Degree of Freedom
DoE	Design of Experiment
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DSR	Design Science Research
EAS	Evolvable Assembly System
FAS	Flexible Assembly System
FBP	Favoured possible Base Placement
FMS	Flexible Manufacturing System
GUI	General User Interface
IEEE	INstitute of Electrical and Electronics Engineers
IK	Inverse Kinematics
IoP	Internet of Production
IoT	Internet of Things
LMAS	Line-less Mobile Assembly System
MaRCO	Manufacturing Resource Capability Ontology
MASON	MAnufacturing's Semantics Ontology
MRS	Multi-Robot System
NIST	National Institute of Standards and Technology
OPTICS	Ordering Points to Identify the Clustering Structure
OWL	Web Ontology Language
PMK	Perception and Manipulation Knowledge

PRISMA	Preferred Reporting Items for Systematic review and Meta-Analysis
PROSA	Product, Resource, Order, Staff Architecture
PPR	Product, Process and Resource
PUMA	Programmable Universal Machine for Assembly
RAMI4.0	Referenzarchitekturmodell der Industrie 4.0
RAS	Reconfigurable Assembly System
RDF	Robot Description Framework
RF	Robot Flange
RMS	Reconfigurable Manufacturing System
ROS	Robot Operating System
SCARA	Selective Compliance Assembly Robot Arm
SOP	Standard Operating Procedures
SPS	Swarm Production System
SPARQL	SPARQL Protocol and Robot Description Framework (RDF) Query Language
SOIL	SensOr Interfacing Language
SUMO	Suggested Upper Merged Ontology
TCP	Tool Centre Point
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language
URDF	Unified Robot Description Format
VDI	Verein Deutscher Ingenieure e.V.
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
WZL	Werkzeugmaschinenlabor der RWTH Aachen

V List of symbols

A	Objective function for K-Means clustering
$a(e)$	Average distance between pose e and all other poses in the same cluster
$b(e)$	Average distance from pose e to all poses not in the same cluster
C	Silhouette coefficient
d	Data points
D	Dexterity index
D_N	Distributed reachability index
D_{RI}	Reachability index
E	Number of reachable poses per voxel
e	Task pose
F	Number of reachable orientations per position
G	Workspace evaluation measure
g	Data point
I	Number of voxels per workspace
J	Jacobian matrix
K	Number of centroids
M	Number of tested orientations per position
m	Dimension of the task
N	Number of tested poses per voxel
N_{rv}	Number of searched reachability vector combinations
p	Poses per voxel
p_{ma}	Performance metrics average
p_{mv}	Performance metric value
q	Joint angle configuration
q_{max}	Maximum joint limit

q_{min}	Minimum joint limit
R	Maximum robot reach
r	Radius of the region of feasible robot flange poses
$r_{l,k}$	Binary indicator indicating assignment of the data point p_l to the cluster K
S	Score matrix containing all valid combinations of all score vectors of the clustering
V	Volume of reachable workspace
W	Weight vector to evaluate valid combinations of base placements
w	Manipulability measure
α	Roll
β	Pitch
γ	Yaw
Δ	Resolution of the robot workspace
ϵ	Data point distance
ε	Element of
ζ	Increments of the golden angle
θ	Rotation around the x-axis of the Tool Centre Point (TCP)
κ	Condition number
μ	Cluster centroids
σ_n	Singular values of the Jacobian
Φ	Rotation around the y-axis of the TCP
ϕ	Golden angle
φ	Rotation around the z-axis of the TCP

1 Motivation, research methodology and structure

1.1 Need for formation planning of mobile robots in changeable assembly systems

Volatile global and regional economic trends challenge manufacturing companies to increase the resilience of their production to remain viable [DOLG22]. Companies must respond quickly to increasing product variety and shortened product life cycles [AREN22, p.1]. Moreover, geopolitical catastrophes like armed conflict [KORN22, p.1] or pandemics [CHOW21, p.21–22] cause disruptions in demand and supply networks, thus enhancing the demand for resilient production. As fixed-coupled production lines cannot provide the necessary resilience, changeable and thus resilient production systems are required to respond to these volatile circumstances [DOLG22; BREC17], providing changeability and adaptability on a physical as well as a logical level [WIEN07; ELMA16].

As one of the main cost drivers in manufacturing, the pressure on the assembly to adapt quickly towards a customer-oriented, individualised economy and market structure, resulting in shorter response times and complex adaptations, has increased [WIEN04; LOTT12, p. 3; HU13]. Studies of WIENDAHL ET AL. (2007) and KLUGE ET AL. (2011) showed flexibility and changeability of assembly systems to be key enablers to staying competitive in the market [KLUG11, p.11; WIEN07], as conventional, fixed-coupled assembly systems are reaching their limits to comply with those requirements [ASAD16; HÜTT21, p.2].

In changeable assembly systems, the complexity of planning and control increases due to the higher degrees of freedom in robot configuration [TOLA22], task allocation [CHEN19b, p.1], and (re-)placement of the robots [ROLD21, p.1]. The demand for more changeable assembly cannot be answered by traditional and flexible manual labour [ELMA21; WANG18] due to high costs and the low availability of trained workers [HENR22; STAT22; COX20; 22a; COX22]. Consequently, automation, automated changeability, and a profound comprehension of intricate manufacturing procedures are emerging as pivotal factors [SHAR22b; AREN22, p.1] to reach the goal of mass customisation while maintaining the benefits of mass production [HU13]. Whilst conventional manufacturing and assembly systems have achieved a significant level of automation, the automated processes are not easily adapted or reconfigured [PEDE16, p.2].

Several solution strategies and paradigms have been proposed to provide the required changeability of assembly systems. Initial studies, such as the one conducted by UEDA (1992) on Biological Manufacturing Systems (BMS), demonstrated the feasibility of a changeable, line-less assembly approach. This research was the first to reveal the potential for mobilising machining and assembly robots. Since then, significant progress has been made in both research and industry. Mobile transportation platforms, primarily Automated Guided Vehicles (AGVs), and automated warehouses have successfully integrated into industrial settings [GEIH20, p.5]. Simultaneously, ongoing research is exploring innovative methods for autonomous navigation and

control of mobile robots in assembly environments, as evidenced by recent studies [CHEN22; AZAD19].

Similarly, on a structural level, different organisation paradigms to organise assembly systems have developed. Following the trend towards more flexibility, assembly paradigms exhibit increasing levels of flexibility. Established methods such as the Flexible Assembly System (FAS), Dynamically Interconnected Assembly System (DIAS), Reconfigurable Assembly System (RAS) or matrix production offer increasing adaptability to external market forces by increasing the amount of reconfigurable and mobile assembly robots [ELMA13, p.642]. As the planning premises often change between design freeze and implementation of the assembly, a general changeability has to be included in the design [FECH18, p.39]. Emerging paradigms, such as the Line-less Mobile Assembly System (LMAS) or the Swarm Production System (SPS), enable the rapid, temporary creation of assembly stations within the factory facilitated by mobile robotic platforms operating in short spatial and temporal cycles [HÜTT21, p.60; SCHO22; BUCK23, p.3].

The automated integration of robots into assembly processes enables flexible and changeable assembly lines capable of coping with fluctuations in supply and demand [ROLD22, p.1]. Due to the rising number of autonomous robots in industrial applications [GEIH20, p.1–2; 22b; SHAR22a], the need for automated configuration, formation planning, and programming of robots increases [BUTT20, p.7; CORV22a, p.472; RODR21, p.49]. The main barrier to employing mobile robots to adapt to dynamic changes remains the substantial manual setup, programming and control expenses [CHEN19a, p.1–2; PERZ19, p.1; MALH22; KALA19], increasing the demand to automate to mitigate the expenses.

Automating changeable assembly systems requires decision support systems at various levels [MAKR12, p.347; FECH18, p.40] to enable the utilisation of assembly robots that can be mobilised and, therefore, placed at new positions in an assembly station to execute their tasks [SCHO22]. This includes job routing, assembly station placement, task-to-robot allocation, robot positioning in relation to the product and tasks, and execution of assembly operations, as well as robot navigation and control [CHEN19a, p.291; KLUG23d, p.7; BUCK23, p.4]. While the seemingly infinite number of possible formations renders the planning and operation of assembly stations complex, it also allows for considerable adaptability to a given assembly scenario. An essential enabler is the evaluation of the spatial location (formation) of mobile assembly robots concerning their capability to execute the assembly tasks [MAKH18; HASS18, p.2; HU13, p.7] to identify formations for fast-changing tasks. The assembly efficiency is determined by those formations, specifying the spatial placement and orientation of all individual assembly robots [LIM17, p.88; CHEN19a, p.1–2; KAWA22, p.1]. While long-term planning allows a tedious simulation and manual variation of the robot base placements until executability is ensured, this is too time-consuming for changeable assembly systems with ever-changing formations [FECH18, p.41]. **Therefore, the need to automate base placement and formation planning of mobile robots in assembly stations arises.**

These conditions give rise to the objective to scientifically investigate the current challenge of planning the formation of mobile robots in assembly stations. The primary purpose is to address the challenges associated with modifying or integrating physical resources, technologies,

or processes within the workflow of assembly stations. Formation planning requires defining the temporary spatial placement of assembly robots for effective assembly operation. This thesis focuses on researching three building blocks of formation planning: firstly, the **semantic modelling and description** of assembly robots and their capabilities to execute a given task; secondly, the **evaluation of executability** of tasks through robot workspaces under consideration of equipped end-effectors; and thirdly, the spatial **base placement of robots** in a formation, ensuring executability of all tasks. These are instrumental aspects for placing robots in a formation that facilitates adaptive assembly operations. Encapsulating the thesis's objective, the following primary research question is investigated: ***Can capability-based formation planning of mobile assembly robots in stations of changeable assembly systems ensure the executability of the assembly tasks?***

1.2 Research objective and application domain

The central research focus of this thesis is the formation planning of mobile robots in assembly stations to find a spatial arrangement or layout (the formation) of the robots to allow the execution of given assembly tasks. This addresses manufacturing companies' business needs and challenges, as outlined above. A reference frame of the research area is provided below by delimiting the research scope, following KUBICEK's research methodology. The theoretical perspective to grasp the research artifact — formation planning — is summarised. The author's understanding of it and the surrounding research area is specified, as visualised in figure 1.1. Hereby, the deeper the blue, the more relevant the highlighted topic. The problem thus outlined is specified by research questions (cf. section 3.4). [KUBI77, p.17–18]

The research artifact of this thesis is placed in the domain of assembly, specifically the domain of changeable assembly systems, characterised by mobile assembly robots. It deals explicitly with industrial assembly while being mindful of its connections to other enterprise disciplines, such as machining, logistics, production engineering, and product design. The assembly organisation forms of changeable assembly aim at small series to single product production since the changeable nature is most beneficial for these products, whilst mass production favours established line production and assembly [HÜTT21, p.12; BUCK23, p.4]. Regarding the considered robot application fields, the focus lies on robotics in assembly whilst considering robot design as a knowledge base regarding the required metrics assessing the robot's capabilities and performance. Adjacent thematic fields following SICILIANO ET AL. (2016), like moving, robot-human interaction or sensors, are noted but not investigated [SICI16]. The scale of the research object is restricted to separate assembly stations, whereby the interfaces to factory planning, robot control, and process execution are taken into account and included (as visualised in figure 2.1). Given the selected scale of the assembly station and the organisational form of changeable assembly, a time frame of a few hours to a few minutes results in which changes occur and new formations have to be planned.

This thesis aims to close the gap between high-level factory configuration planning and the low-level process execution and control of mobile robots. The overall objective of this work is a formation planning of mobile assembly robots, consisting of an allocation of assembly tasks to mobile assembly robots, under consideration of their spatial capabilities as well as the spatial

Disciplines	Product development		Machining	Assembly	Logistics	
Assembly organisation form	Dedicated assembly line	Reconfigurable Manufacturing Systems (RMS)	Dynamically Interconnected Assembly Systems (DIAS)	Changeable assembly	Line-less Mobile Assembly Systems	
Production volume	Single product production		Small series production	Series production	Mass production	
Robotics [SICI16]	Robots & humans	Robots in application	Sensing & perception	Robot design	Movement & navigation	Manipulation & interfaces
Scale [SCHO07]	System (here: entire factory)		Subsystem (here: assembly station)	Machine (here: combined resources)	Component	
Considered time frame	Weekly	Dayly	Hourly	Every minute	Every second	Every millisecond

Figure 1.1: Heuristic reference frame relating formation planning to neighbouring domains.

placement of the robots, ensuring the executability of the allocated assembly tasks. Therefore, the following building blocks are needed:

Description model of mobile robots and their capabilities: To realise formation planning, a suitable description of the object of research of the formation, the assembly tasks, the robots, their components and their capabilities are fundamental. Based on this description, the tasks must be allocated to combined robots (consisting of their components: a mobile base, a robot arm and an equipped end-effector) considering the resulting capabilities exhibited by the combination of the robot components.

Executability evaluation: To evaluate the formations in terms of their feasibility in executing the assembly operation, the executability of tasks in the robot's workspace has to be investigated. In dependence on the tasks at hand, a method to choose relevant measures to evaluate the executability of this specific task has to be developed. Depending on the selected executability evaluation measures, the workspace of the robot has to be evaluated. When evaluating the workspace, both the robot — combined of the robot arm and the end-effector — and the type of assembly task must be considered to integrate all spatial influences on the executability.

Formation planning through task allocation and base placement: A robot base placement method is to be formulated to determine the formation of the assembly robots in dependence on the spatial distribution of the assembly tasks. Therefore, the distributed tasks have to be clustered and assigned to a single robot at a single base placement. By applying the workspace evaluation of the robot, the executability of all assembly tasks has to be ensured. Finally, the performance is to be validated, and applicability is to be verified.

The research conducted in this thesis was partially developed as part of the "Future Assembly" workstream of the research project IoP¹. Figure 1.2 visualises the central artifacts as a goal scenario and includes the artifacts into the (abstracted) layers of the semantic model of the IoP as presented by BRECHER ET AL. (2017, p.121). The IoP is a practical implementation of the

¹Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — EXC–2023 Internet of Production — 390621612

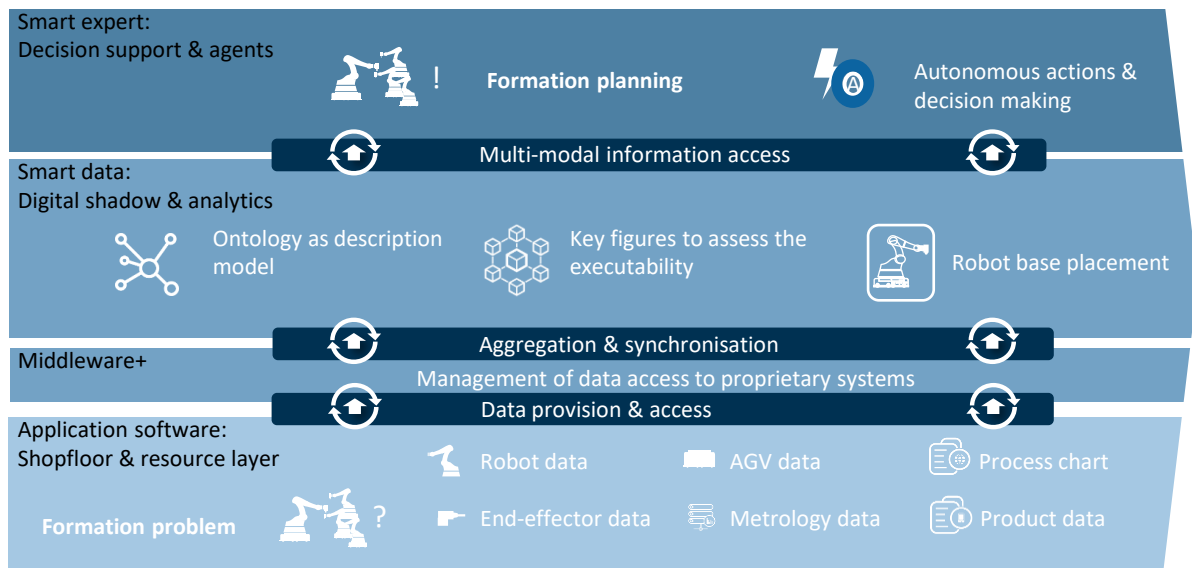


Figure 1.2: Formation planning and its building blocks of description model, executability evaluation and base placement of mobile robots in the semantic model of the Internet of Production (IoP).

Internet of Things (IoT) concept. It aims to horizontally connect cyber-physical devices and utilise field data to generate value-added production services. The goal is an inter-operative, cross-domain network of stakeholders in the production process, enabling real-time control of production, data storing and processing, and secure data exchange. [PENN19b; PENN19a; LIEB21]

The researched formation problem originates on the shop floor and robot level in the initial system layer. Based on the input of process charts, a formation of robots capable of assembling the requested product is to be derived. The subsequent layer provides middleware managing access to proprietary interfaces and supports communication protocols. The third layer transfers aggregated and synchronised data into a digital shadow. Here, the mobile assembly robots and their capabilities are described and simulated. This layer facilitates analysing and evaluating the current state of the digital shadow (e.g. in ontologies) based on complex simulations (e.g. in Robot Operating System (ROS)), cluster algorithms, heuristics, measures and key figures. The ultimate objective of the system's semantic layer is to close feedback loops based on the acquired and processed data. This involves automated decisions like process reconfiguration or planning mobile assembly robot formations, resulting in autonomous adaptations of the assembly station. Formation planning integrates into the context of the IoP, contributing to automated decision-making for real-time control of mobile assembly robots.

1.3 Research methodology and derived structure

To ensure the relevance and focus of the research, it is advisable to proceed in a structured manner according to established research methods. [MARC95, p.1] Research in the sense of the systematic and comprehensible recording and analysis of operational reality represents an indispensable complex of activities for achieving research goals [KUBI77, p.5].

Due to the practice-driven relevance of the research on the formation planning of mobile robots in assembly, it classifies as application-oriented research with the aim of developing a substantive solution for a practice problem. To follow ULRICH (1981) and generate relevant and practically useful knowledge through conceptual research, research should not stem from observations and theories but rather from real problems. [ULRI81, p.5, 11] According to KUBICEK (1977), the research goal for applied research is increased understanding and the resulting possible mastery of reality. [KUBI77, p.7] By applying the research methodology methods, the practical concerns of formation planning can be scientifically investigated [ULRI81, p.5].

As this thesis falls within the field of design-oriented sciences, the research methodology employed is the Design Science Research (DSR) method, as proposed by HEVNER ET AL. [HEVN03; HEVN04; HEVN07]. The goal of the design-oriented sciences is to develop knowledge for designing solutions to problems in their field of application, in contrast to the explanatory sciences, whose goal is to understand their interrelationships [VAN 05, p.20]. The research here falls in the field of design theory since it aims at utilising computational power to aid with planning and controlling in assembly, which applies the general idea of design theory to an explicit problem [SIMO96, p.114]. The research topic of formation planning of mobile robots in assembly is empiric and pragmatic (according to [KUBI77]), applied (according to [ULRI81]), and falls in the area of design science and information systems (according to [MARC95]). Therefore, it shall be investigated following the proposed research approach of DSR of HEVNER (2007).

Design science is rooted in engineering and the sciences of the artifacts and aims at creating and researching *design artifacts* [SIMO96, p.111; HEVN03]. HEVNER ET AL. and MARCH AND SMITH classify design artifacts depending on their nature into constructs, models, methods and instantiations [MARC95; HEVN04], whilst IIVARI proposes to differentiate artifacts according to their field of application [IIVA07, p.5] The DSR as proposed by HEVNER AND MARCH in 2003 (here called "Information systems research framework", which was later changed to DSR), provides a synergistic approach by connecting *Research* with its contextual *Environment* and domain *Knowledge base* [HEVN03].

DSR provides a guideline for design science on how to proceed, divided into three iterative research cycles. The original DSR framework is presented in the appendix in figure A.1 while the applied version is pictured in figure 1.3, both showcasing the three domains and their connection by the three activity cycles. In the relevance cycle, the requirements from the application domain (the environment) are derived, and the criteria to evaluate the applicability of the artifacts are formulated. In the rigor cycle, the existing knowledge base and its artifacts and processes are investigated, the relation of the new research to existing research is highlighted, and applicable artifacts, measures and fundamentals are discussed. The central element of the research — the development of new and original artifacts — takes place in the design cycle, aiming to fulfil the requirements identified in the relevance cycle by incorporating the relevant existing research identified in the Rigour Cycle with new methods. A characteristic element of the design cycle is the iterative development and integration of knowledge gained from theory and practice, which was also postulated as necessary by ULRICH (1981). After several iterations, the entirety of the artifacts is tested for applicability in practice, whereby the

design cycle transitions into the relevance cycle. Whilst the cycles of relevance, research and rigour were already sketched out by HEVNER ET AL. in (2003), they were explicitly discussed as necessary research activities in 2007. [HEVN07; HEVN03] The structure of the thesis can be mapped to the structure of DSR, as presented in figure 1.3.

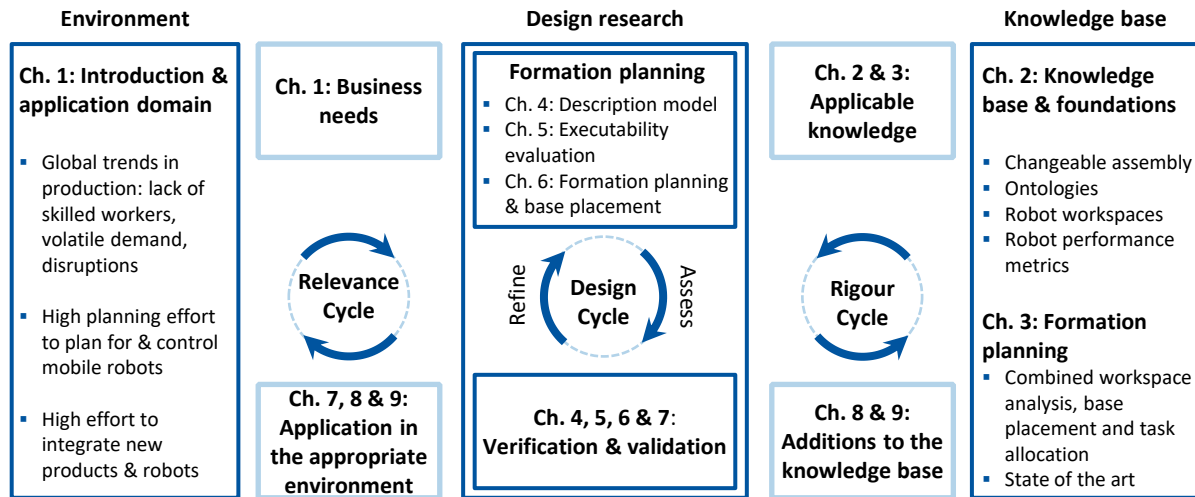


Figure 1.3: Structure of the thesis according to Design Science Research.

Highlighting the system development process, the thesis structure is additionally mapped to the V-model in figure 1.4. The V-model is an established process model for developing mechatronic, information, and cyber-physical systems. It interconnects the respective tasks during development. The main development phases are differentiated into artifact decomposition through the elicitation of requirements, model design, and the system architecture design, as well as implementation and integration. Parallel to the main development phases are verification and validation integrated into the model as connecting elements between the two legs of the V. [VERE21, p.19-37] Whilst figure 1.4 indicates the mapping of the overall thesis structure into the V-model, each design cycle for the research of each respective partial research artifact necessitates conducting a similar structured V-model-based development process.

Figure 1.5 specifies the resulting structure of the thesis, derived from combining the DSR approach with the V-model. The respective design cycles and verification and validation phases per artifact are specified. Whilst the enclosing chapters 1, 3, 7, 8, and 9 encompass the entire research area of formation planning and detail the requirements and resulting validation of those requirements in application, the respective main chapters 4 – 6 contain individual cycles of the artifact development process from requirement until the fourth step of artifact verification. The overall validation of formation planning is concluded in chapter 7.

Chapter 1 motivates the need for automated formation planning from the application domain, defines the subsequent goal to evaluate the feasibility of formations of mobile robots in assembly, and outlines the structure derived from the research methodology. The application domain and the founding knowledge base are detailed in chapter 2. Chapter 3 reviews related work in the state of the art. Based on the review, research gaps are identified and transferred into the underlying research questions in section 3.4. Chapters 4 – 6 provide individual details on the artifacts that address the underlying research questions. In the **first major design cycle**,

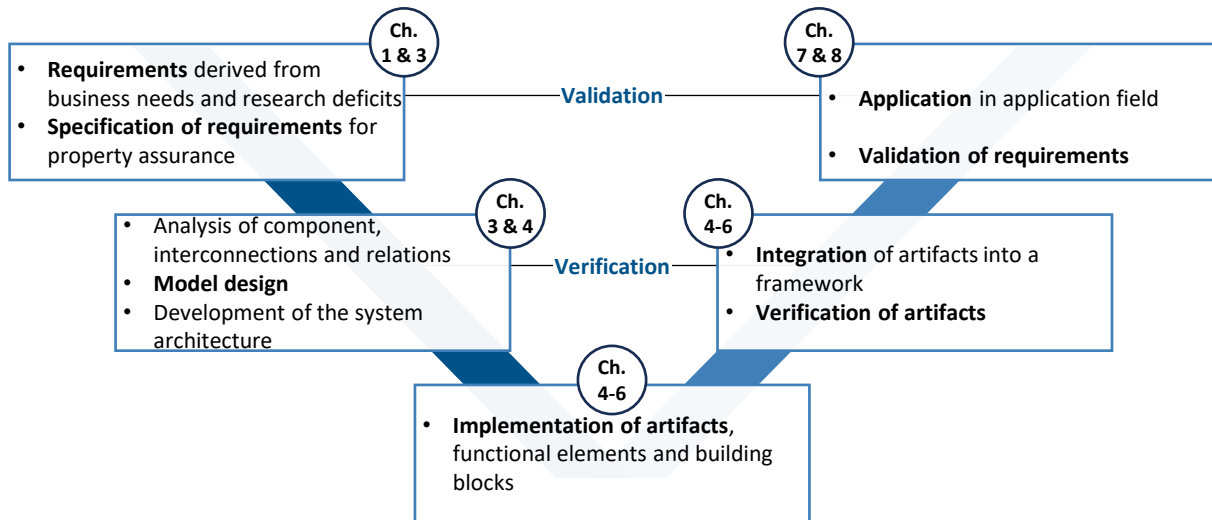


Figure 1.4: Research artifact development process, following the V-model, based on [VERE21].

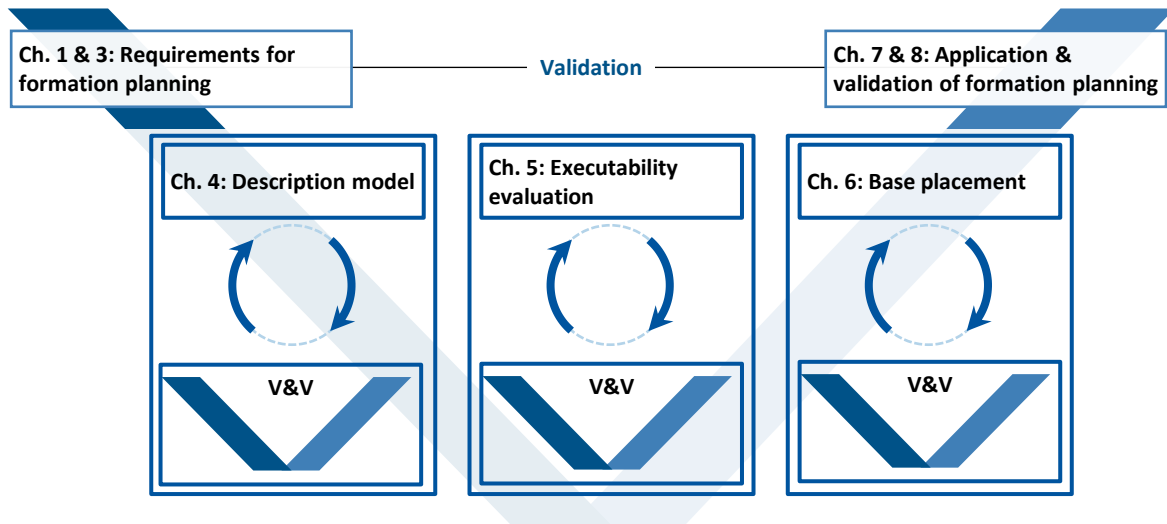


Figure 1.5: Chronological thesis structure, derived from the combination of the Design Science Research and the V-model development process.

presented in chapter 4, a **description model** is developed as the basis for the subsequent steps. The artifact of the description model was adapted chronologically in the course of the work to progressive results and thus reflects the iterative character of the DSR. The literature review to identify characteristic properties of robots presented in the knowledge base (see section 2.3) is leveraged to establish a capability model in the ontology. The **second major design cycle**, presented in chapter 5, explores the second research question, resulting in the artifact of **executability evaluation** of the robot's capabilities to execute tasks in its workspace. The focus is on implementing the aforementioned semantic capability model and extending it to a spatial expression of the robot's capabilities in its workspace, including dependence on different metrics, tasks, and robots and their end-effectors. In the **third major design cycle**, presented in chapter 6, the previous considerations are consolidated into a **formation planning and base placement**. This chapter introduces a clustering-based method of grouping distributed tasks into robot-workspace-sized clusters. For each cluster, a base placement is found. The

resulting formation — consisting of the base placements and allocated tasks — is assessed for executability. Finally, the artifact of formation planning is verified, validated and applied in chapter 7. A critical reflection and discussion of the designed artifacts and their relevance to their application domain are carried out in chapter 8. Finally, the findings and further research directions are summarised in chapter 9.

2 Mobile robots in changeable assembly systems: Knowledge base and foundations

Following the DSR, the first step of researching the formation planning of mobile assembly robots is to consult the knowledge base, thereby ensuring rigour of the research [HEVN04, p.80; HEVN07, p.18]. This chapter describes the knowledge base and foundations in the research area, presents applied methods, clarifies the terminology, and identifies the need for a systematic literature review on formation planning, which is presented in chapter 3. Section 2.1 deepens the delimitation of the research area presented in figure 1.1, and defines the terminology of assembly and robotic systems. Previous research in robot workspace evaluation, along with the associated areas of performance metrics and base placement, is summarised in section 2.2, highlighting the need for further research on formation planning. Section 2.3 overviews description models and semantic modelling of assembly robots as the foundation for developing an ontology in chapter 4. Section 2.4 summarises task clustering algorithms, as the foundation for the base placement conducted in chapter 6. Finally, section 2.5 details the applied methodology throughout the verification and validation of the research artifacts.

2.1 Changeable assembly systems

The following section 2.1.1 details the scope of the investigated research area, provides an overview of the changeability terminology employed, and relates the research artifact of formation planning to adjacent research. The terminology concerning mobile assembly robots and their components is defined in section 2.1.2.

2.1.1 History and terminology of changeable assembly

Due to heightened global competition, manufacturing companies continuously encounter more frequent and unpredictable market fluctuations [KORE10; HU11, p.718; ELMA21]. Additionally, the demand for customised or personalised products increases the diversity of products [HATV78; KORE99, p.527; FERR12, p.2]. Therefore, production and assembly systems developed from mass production characterised by a high volume of similar products [KORE99, p.527] organised in **Dedicated Manufacturing Systems (DMSs)**² to more flexible and changeable³ systems: **Flexible Manufacturing Systems (FMSs)**⁴ and **Reconfigurable**

²DMS are characterised by transformability: the ability to switch production within different groups and families and agility: the ability to react to market volatility [WIEN07]

³The term changeability encompasses various paradigms and is defined as the ability to efficiently and proactively adapt the structures and processes of the factory at all levels in response to changing stimuli [WIEN07; ELMA09, p.10].

⁴Flexibility: the ability to switch production within a part family [TERK09, p.7–9; ELMA09, p.4; ELMA05, p.2; SETH90; BROW84]),

Manufacturing Systems (RMSs)⁵ [FERR12, p.2]. FMSs increase the diversity of products by employing programmable, multi-purpose resources [KORE99, p.527; SCHO07, p.714; LOTT06; BROW84]. In RMSs, production capacity and functionality can be adapted within a product family in response to external influences [KORE99, p.527–529; ELMA05; SCHO07, p.714; HU13, p.6]. Here, only systems exhibiting reconfigurability or changeoverability⁶ are considered of interest, as they are likely to incorporate mobile assembly resources.

Continuing the trend of increased production and therefore assembly flexibility, **Evolvable Assembly Systems (EASs)** enable "plug and produce" based on inter-connectable, simple, and reconfigurable task-specific elements ("system modules") [FERR12, p.1; ONOR10]. The first attempt to adhere to dynamic changes in the internal and external environment production by mobilising assembly resources can be found in **Biological Manufacturing Systems (BMSs)** [UEDA01, p.319–322; UEDA92; KATA19]. Despite BMSs showcasing the benefit of mobilising resources, the formation planning methods remain rudimentary, lacking resource coordination and in-depth capability analysis.

The latest developments in the emerging trend of increasing flexibility of production and assembly systems are **Line-less Mobile Assembly Systems (LMASs)**⁷ and **Swarm Production Systems (SPSs)**. Both share the concept of line-less⁸ production and assembly, resulting in highly adaptable systems exhibiting customised order routes [SCHO22]. Formation planning of mobile robots is pertinent in SPS and LMAS featuring mobile robots that undergo regular location changes. The three central principles are a clean floor approach, mobilised, multi-purpose robots and dynamic process and job routes [HÜTT19, p.725–726; HÜTT21, p.60; BUCK23, p.18–20]. Similarly, SPSs promote a dynamic production "composed of multiple, different purpose autonomous mobile entities" [SCHO22, p.108], thereby enabling dynamic routing and adaptive reconfiguration. In difference to LMAS SPS comprises the entire production, as opposed to LMAS focusing on the assembly. LMASs and SPSs are assembly and production paradigms whose realisation necessitates the formation planning of mobile robots to enable operation. Therefore, they are the direct application field of the thesis.

2.1.2 Mobile robots in changeable assembly

As many terms relevant to assembly and robotics do not have a generally accepted definition, they are treated and delimited in the following section.

An **assembly system** is an industrial facility designed to receive individual parts and cohesively integrate (assemble) them into products or groups of higher production order [FREI08, p.98; VDI-78, p.3; CIRP20, p.1; DEUT22, p.5]. Here, the three levels in the assembly system of factory, station and robot — as visualised in figure 2.1 — are considered. The macroscopic level of factory configuration planning, the mesoscopic level of station formation planning and

⁵Reconfigurability: the ability to switch production within a part family through reconfiguration of physical resources [ELMA05, p.2; ONOR06, p.325; ELMA09, p.4]

⁶Changeover-ability: the ability of a production or assembly system to perform a known operation at any time [WIEN07].

⁷originally promoted as Free Float Flawless Assembly (F3A) [BREC17, p.339–368]

⁸Two categories of systems are regarded as line-less: matrix systems, with fixed assembly stations and the focus here: systems where all assembly robots are mobilised, e.g. LMAS or SPS. [BUCK23, p.23; HÜTT21, p.21]

the microscopic level of process execution and control are differentiated. At the highest level of abstraction, the factory comprises the temporary locations of the assembly stations and their allocated tasks. An **assembly station** is considered a set of individual assembly robots in a defined factory area connected by a common goal: to execute a given number of assembly tasks. The **station configuration** is determined by the number and type of the resources and the designated area. A changed set of resources results in a new configuration.

The integration of formation planning between factory planning and robot control presupposes existing procedures for the prior allocation of assembly robots and tasks to the assembly stations, as well as the subsequent trajectory planning and control of the respective mobile robots. While there are several studies and procedures for the two enclosing levels (e.g. [BUCK23; VAN 98; LEIT06; HOFF17] and [LIND05; LAVA06; PAND17; ZHU21; ROLD22; KLUG23d, p.9–10] respectively), formation planning at station level has so far hardly been addressed.

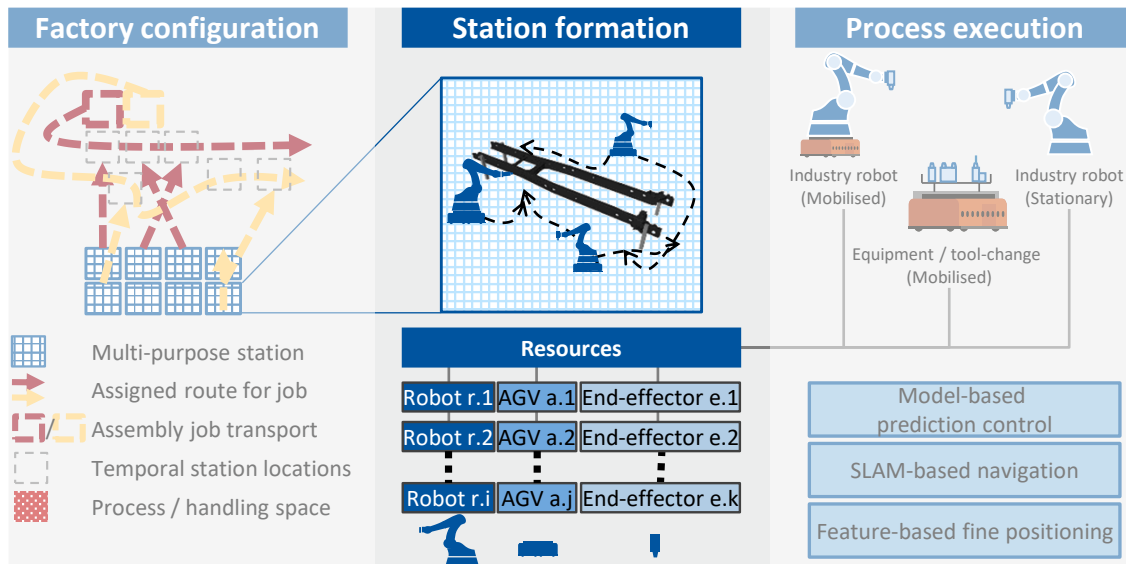


Figure 2.1: Hierarchic abstraction levels of operation planning and control in changeable assembly systems.

The **station formation** refers to the temporary spatial arrangement of specific robots and allocation of tasks to those robots in an assembly station. Figure 3.2 visualises a formation in an assembly station, including the colour-indicated allocation of task poses to robots to specific base placements. The precondition for a feasible formation is the executability of the set of allocated assembly tasks by the robots at their assigned locations / base placements. The following terms are utilised synonymously to the above understanding of "formation": reconfiguration, layout, relocation of resources, and arrangement of resources [HÜTT21, p.61; HU11, p.718; SPIC02, p.2].

Following the terminology established for LMAS, a **physical resource** is defined as a physical object related to the assembly process by either executing an assembly task or being part of the product [BUCK19b, p.229]. Here, the term resource encompasses all assembly resources, like sensors, metrology systems, robots, human workers, or tool-changing systems. Multiple robots with redundant capabilities create a flexibility corridor for allocating tasks to robots

within assembly planning, thereby enhancing the effectiveness, efficiency and fault tolerance [ROLD21, p.1]. Employing teams of heterogeneous robots is assumed to increase the process efficiency [ROLD21, p.1].

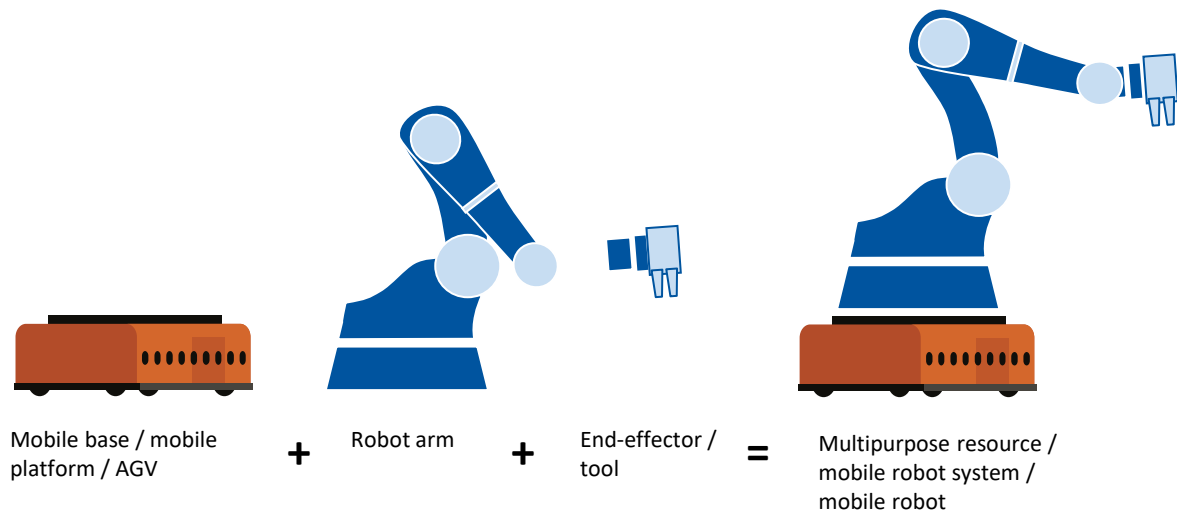


Figure 2.2: Components of a mobile multipurpose robot, based on [TOLA22; STAN21; ZUMP23].

Here, the focus lies on mobilised **robot arms** equipped with an end-effector as the executing and value-adding resources in assembly. An industrial robot is defined as an automatically controlled, reprogrammable mechanical resource adaptable to multiple purposes through physical changes, e.g. by changing the end-effector [OLIV19, p.12–13; STAN21, p.1, 2; KUNZ11, p.3]. The mobile multipurpose robots considered here include a robot arm, a mobile base, an end-effector and internal and external sensors, as depicted in figure 2.2. Depending on the combinations of parts in the mobile robot, it can have multiple purposes. Multipurpose robots achieve redundancy in operation. Here, only industrial six-axis robot arms with six revolute joints (Programmable Universal Machine for Assembly (PUMA) robots) are considered.

Robot **autonomy** refers to the ability of a robot to control itself and to execute actions without external control. The autonomy of robots constitutes sensing, planning and acting based on the sensory input to reach a goal. [BEER14, p.3; SCHO07, p.712; OLIV19, p.5; VERE18, p.4]

To execute tasks, the robot is equipped with an **end-effector** onto the robot's flange. An end-effector is a specialised device designed to be attached to a robot with a hardware and software interface. Its purpose is to enable the robot to perform its allocated task by moving in relation to and transferring energy to the workpiece to alter its form and pose. [STAN21, p.4; DEUT22, p.5]

The objective of utilising robots in assembly is the execution of **assembly tasks** (synonyms found in the literature are: operations, job) [SPIC02; ABDO17; ZHAN19c]. Here, an assembly task is defined as a six-dimensional pose in space where a specific assembly operation is to be executed. Thereby the pose constitutes a position and an orientation in space [STAN21, p.7].

Whether a robot can execute a task depends on the spatial relation between the task and robot as well as the robot's **capabilities**. Capabilities encapsulate the ability of a robot to

execute or perform a task, more generally, to cause an effect in a specific context. Capabilities (sometimes: "skills") encapsulate reusable functionality, enabling the connection of Product, Process and Resource (PPR) and modular and efficient control within the system. [GINS22, p.9; ONOR06, p.327; KLUG11, p.77] A detailed discussion of the term capability can be found in section 2.3.2.

The robot can interact with its surroundings in its **workspace**. The workspace can generally be defined as the space a reference point can reach [STAN21, p.8; FREI08, p.98]. The different definitions of a robot's workspace and the way of deriving those are discussed in section 2.2.

2.2 Workspace evaluation of mobile robots in assembly

Evaluating the individual robot workspace generating knowledge about the area where the robot can execute tasks is fundamental to evaluating robot base placements. Based on quantitative metrics, the feasibility of the robot to execute a task can be compared, a precondition for automation. The following section overviews the state-of-the-art workspace evaluation and performance metrics. At first, a definition of workspace types is given (section 2.2.1), followed by an overview of methods to compute workspaces. Next, established performance metrics are compared based on the overview presented in [KLUG22d] (section 2.2.2). Section 2.2.3 analyses existing libraries and methods computing different robot metrics and workspaces. This section summarises how metrics can be applied for the base placement of mobile robots in section 2.2.4.

2.2.1 Robot workspaces

Placing mobile robots in assembly stations to execute tasks raises the question of how to describe where the robot can execute tasks with which characteristics. The following discusses how to describe a robot's workspace, which is the volume expanding from the robot's base in which the robot can execute tasks.

Reachability and dexterity of robot workspaces were researched firstly by KUMAR ET AL. in (1981). KUMAR ET AL. (1981) put forth the concept of the connection between kinematic geometry and manipulator performances. GUPTA ET AL. (1982) conducted a thorough analysis of the impact of hand size on the workspace. Analysing the robot's workspace has been applied to enhance human control of medical equipment during operation [ZHAN22], placing medical robots in their environment [TRAB22] or to plan the grasping motion of a robot [AKIN18]. Two different approaches to evaluate the workspace can be differentiated: Firstly, as a view on safety during human-robot collaboration ranging from complete collision avoidance to direct collaboration in a shared workspace by evaluating a potential collision's harmfulness, e.g. in a so-called Safety Map [MANS22]. Secondly, to evaluate the capability of the robot to execute tasks in its environment [KUNZ17, p.13; KLUG22d], which will be the focus here.

Definition of a workspace The workspace of a robot comprises the set of all poses or positions reachable under defined boundary conditions in the Cartesian space by the robot flange or TCP⁹.

⁹A TCP is a "point defined for a given application with regard to the mechanical interface coordinate system" [STAN21, p.9].

Reachability depends on the kinematic structure of the robot, the environmental conditions and the relative pose of the workpiece to the robot. Joint limits, the size of the robot segments, as well as the applied characteristics (e.g. requirements of robot flange velocity or applicable force) thereby define the shape of the workspace [KUNZ17, p.vii].

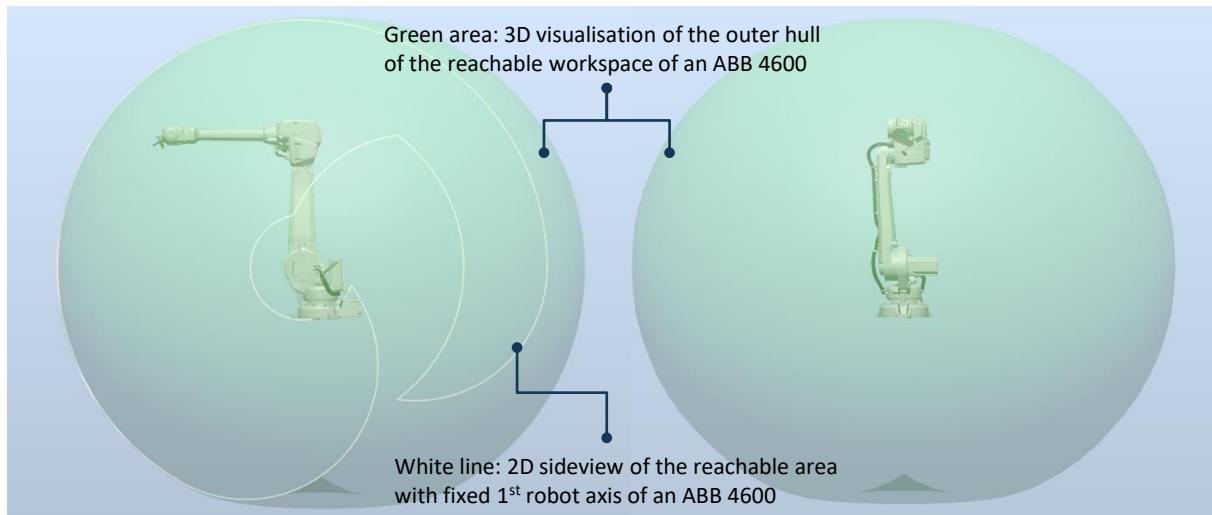


Figure 2.3: Reachable workspace of a "decoupled" 6-axes robot (ABB 4600) with six revolute joints (generated in RobotStudio version 20.1.9014.1).

Provided below are the typically used workspace differentiation based upon [PATE15, p.550; KUNZ17, p.13; DASH05; CAST08, p.15]. Considering physical robots requires extending the theoretically possible workspaces by limitations of the joints, e.g. to account for self-collision, collision with or damage of additional peripheries — like loose wires. [ANGE16, p.401]

- The **reachable workspace** (sometimes "work envelope") is defined as the sum of all positions reachable in at least one orientation by the robot flange as visualised in figure 2.3 [KUMA81; GUPT82]. When considering a six-axis robot with a "decoupled architecture" (the axes of the last three revolute joints intersect in one point) [ROTH75, p.47], the first three joints and segments of the robot position the robot's flange in a specific position (x,y,z). The first three segments and joints build the robot's so-called "regional structure", as visualised in figure 2.4. [ANGE16, p.401]
- The **dexterous workspace** is defined as the sum of all positions to be reached with all orientations (pose $\hat{=}$ position + orientation) by the robot flange [ANGE16, p.404; KUMA81; KUMA17]. Therefore, the "local structure" — the robot's last three joints — extends the regional robot structure to orient the robot flange at a position [ANGE16, p.401].
- The **workspace with constant orientation** is defined as the sum of all positions the robot flange can reach in a specific/constant orientation.
- A **workspace with partial orientation angle** envelops all positions the robot flange can reach with a range of angles less than 360° .
- The **orientation workspace** includes all orientations the robot flange can reach from a specific position.

Figure 2.4 visualises a "decoupled robot": When the rotation axes of the last three revolute joints (towards the free end of the open kinematic chain) intersect at a common point, the orientation and position become independent of each other [ROTH75, p.47]. The first three axes (the main axes) form the "regional structure" and position the robot flange. The last three joints (the minor axes) form the "local structure" ($\hat{=}$ wrist) and orient the robot flange in its position.

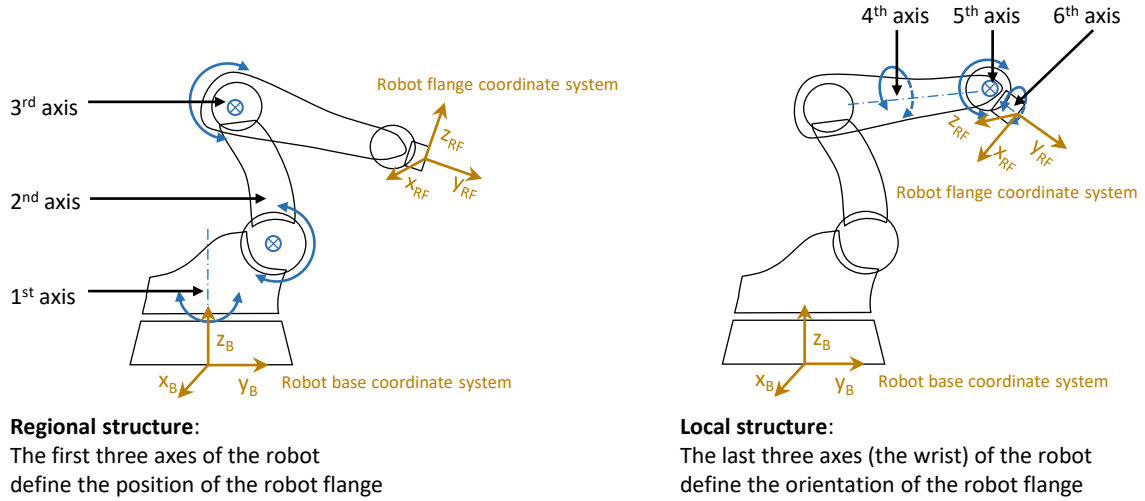


Figure 2.4: Topology of a "decoupled" 6-axes robot.

Generally, if no further differentiation is provided, "workspace" is synonymous with "reachable workspace". The reachable workspace V can be described as shown in equation 2.1, with the robot flange position $x \in \mathbb{R}^3$ in Cartesian space, the joint angle configuration q in joint space, the joint limits q_{min} and q_{max} and the robot forward kinematic $f(): \mathbb{R}^n \rightarrow \mathbb{R}^3$. [ZHAN19b]

$$V = \{x = f(q) | q_{min} \leq q \leq q_{max}\} \quad (2.1)$$

To **compute the workspace**, analytic, geometrical and numerical approaches exist [CAST08, p.15; ZHAN19a, p.3837; DASH05, p.777]. ABDEL-MALEK ET AL. (2006) summarise workspace calculation methods. TSAI ET AL. (1981) derived an **analytical** solution based on loci curves for robot arms with two or three links. Insight into the topology of robot workspaces and typical forms and holes was given by GUPTA ET AL. (1982). Even though analytic methods provide an exact representation of the workspace, they often do not incorporate constraints, like joint limits, and solving the analytic representation can be computationally expensive [CAST08, p.16]. Moreover, analytic methods require specific modelling for every robot [VAHR15, p.785].

BONEV ET AL. (2001) proposed a **geometrical** solution to calculate the workspace of a robot. Starting from the last joint, the area of reachable poses is calculated (e.g. a ring for a revolute joint). Following the kinematic chain backwards for each joint, the area the robot flange can swipe is calculated, as visualised in figure 2.5 [KUNZ17, p.15; DASH05, p.2; BONE01]. Geometrical solutions are aimed at representing the boundaries of the reachable workspace instead of properties or characteristics inside the workspace.

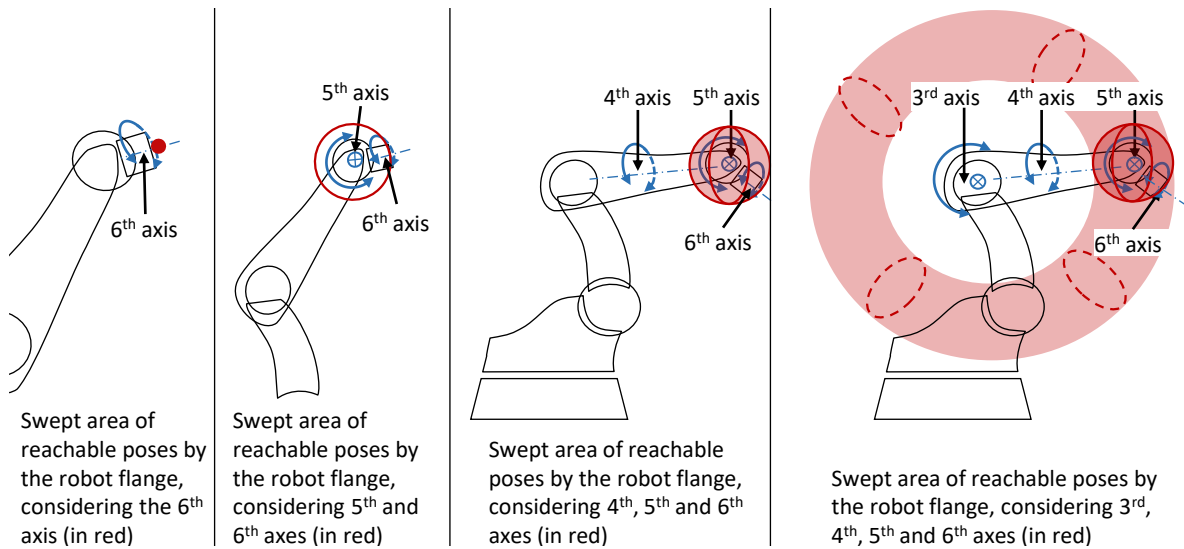


Figure 2.5: The first steps in the geometric approach to determining the reachable workspace, by calculating the reachable/swept area by successively considering one additional axis.

Numerical methods to investigate the reachable and dexterous workspace boundaries have been proposed [VIJA86; TSAI86; DASH05]. Compared to analytical approaches, kinematic constraints can be included [ANGE16, p.404]. To calculate the reachable workspace, a possibility is to sample a combination of joint values through Monte Carlo-based sampling or equal interval method (splitting the range of each joint into equidistant parts) and calculate their respective forward kinematics [ZHAN19b]. Numerical method sample in the configuration space, joint space or Cartesian space. Sampling in Cartesian space faces the disadvantage of closeness to singularities, whilst configuration and joint space do not lead to uniform sampling in the workspace. **Discretising** the reachable workspace equidistantly in its six dimensions and calculating the inverse kinematic solution for each discrete pose provides a Cartesian sampling numerical method, applicable to all robot topologies [BONE01, p.2]. The capability map proposed by ZACHARIAS ET AL. (2007) gives one example to visualise such a discretising method for calculating the reachable workspace (detailed in section 2.2.3).

2.2.2 Metrics to evaluate robot workspaces

Due to its mechanical structure, a robot does not exhibit isotropic performance throughout its workspace. Thus, when employing a robot for a particular purpose, it becomes imperative to validate whether the robot possesses the requisite characteristics. Evaluating the performance of mobile robots requires quantifiable performance metrics [PATE15, p.2]. Initially, robot performance metrics were investigated to design robots [ANGE16; YOSH85a; NATI75]. The same metrics can be applied to assess the performance of a robot in executing a task in its workspace or to plan its base placement accordingly.

The following section overviews key figures, metrics, and performance indicators and analyses their applicability for formation planning in mobile assembly stations. The term metric is defined, followed by a classification scheme to categorise metrics. The overview is structured following three leading sources: the chapter "Design and Performance Evaluation" of the handbook of robotics by ANGELES ET AL. (2016), a review on "Manipulator Performance

Measures" by PATEL ET AL. (2015) and the recent survey on "Metrics for Manipulator Design, Control, and Optimization" by RUSSO (2022). Relevant performance metrics are introduced. An extensive list of performance metrics can be found in table D.1.

Quantitative metrics provide a means to design, analyse and control [CHAN88] by providing a variable to be optimised [PATE15, p.548]. Since research on robot performance evaluation is diverse, manifold definitions and ambiguous terms (e.g. measure, key figures, key performance indicators, index) can be found. Here, the following **definition** (based on [VAN 96, p.1; PATE15, p.2; ZIMM17, p.7; RUSS22, p.3]) is applied: **Performance metrics quantify a robots performance characteristics in its workspace, aiming to describe, analyse, and compare robots.**

The range of existing metrics can be classified according to their properties and considered input. This thesis uses the following classification, established in [KLUG24b]:

- Range: local or global [PATE15, p.549; RUSS22, p.4]
- Physical reference: kinematic or dynamic [PATE15, p.549; RUSS22, p.4–5]
- Task reference: intrinsic or extrinsic [PATE15, p.549; RUSS22, p.4]
- Scale: absolute or relative [RUSS22, p.4]

Global metrics are configuration- and pose-invariant and evaluate the entire workspace in a single value (e.g., maximum reach or maximum payload). Local parameters depend on the robot's configuration and pose (e.g., manipulability measure w or dexterity index [YOSH85b; TANE00]). They represent a local property, which may vary for each pose and offer the possibility to analyse specific workspace areas and configurations during path planning and robot control. [PATE15, p.549] Local metrics are mostly calculated with the Jacobian of the joint angles. [RUSS22, p.4] Most metrics are determined discretely for individual configurations, i.e. angular postures of the individual robot [KLUG22d].

Kinematic and dynamic metrics evaluate the robot's performance in dependence on the physical behaviour. Kinematic metrics, like repeatability, total accuracy or speed, are functions of the robot's structure and are commonly calculated based on the Jacobian. In contrast, dynamic metrics incorporate effects like inertia, forces or torques caused by the dynamic movement of the robot and its payload. [PATE15, p.549; RUSS22, p.5]

Extrinsic metrics compare the requirements to execute the task with the characteristics the robot provides, e.g., the required force transmission during the task trajectory with the robot's capability to provide these forces during operation. Therefore, the exact requirements for task execution have to be known. [PATE15, p.549; RUSS22, p.4] Intrinsic metrics depend on the robot itself: its topology, and mechatronics. Examples of intrinsic metrics are the manipulability measure [YOSH85b], dexterity index [KUMA81; MAKH18], and condition number [YOSH85a, p.8] since they only rely on the robot's structure and configuration. [STUR90; CHIU88]

Their quantified nature renders metrics to be either absolute or dimensionless, relative. Absolute metrics, like speed, velocity or payload, are characterised by their units. Relative metrics do

not depend on a general scale and set a measured characteristic in relation to an absolute or reference. [RUSS22, p.5]

Robots' performance metrics must balance general descriptions and specific details. Generic metrics lose significance, while overly specific ones hinder comparisons. Industry favours intrinsic and global metrics, while researchers prefer extrinsic and local metrics. The field lacks standardisation due to the diverse range of robots and applications. [RUSS22, p.13–14; ZHAN19a, p.3826] Additionally, and if a business viewpoint is of interest, other performance metrics, such as lead time, due date performance, overhead costs or throughput, are of interest [LEIT08, p.4; ELMA09, p.10].

Numerous assembly operations, including tasks like welding [HUO08], composite layup [MALH21], 3D printing, or spray painting [CHEN16], necessitate continuous end-effector paths while adhering to additional execution constraints. In difference, distributed tasks, for example, pick and place tasks [ZHAN23] or grasping tasks [VAHR13], are defined by a small number of distinct, discrete task poses. Whilst some task types, like inspecting or rolling something on, allow for a broader range of acceptable task poses or orientations during execution (as visualised in figure 2.6) [MALH19a, p.5588; TRAB22, p.6], others, like pressing in or painting with defined thickness, are limited to defined task poses (defined by position and orientation).

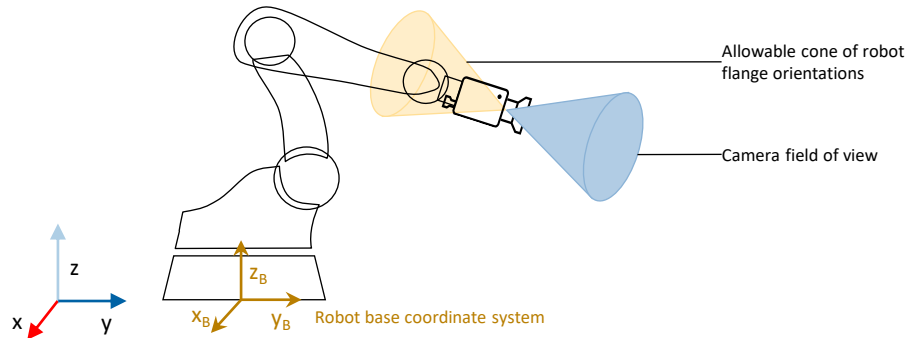


Figure 2.6: Example of the allowable cone of robot flange orientation during visual inspection.

The **manipulability measure**¹⁰ quantifies the achievable velocity from a specific robot configuration [YOSH85b, p.4] and the distance of the configuration to the closest singularity [TANE00, p.6]. The manipulability measure is an established metric for robot path and trajectory planning [ZACH07; VAHR12; ROZO17; KLUG23d; CORV22b; SHAH22]. The manipulability measure is expressed in equation 2.2.

$$w = \sqrt{\det J(\theta) J^T(\theta)} = \sigma_1 * \sigma_2 \dots \sigma_n = \prod_{i=1}^n \sigma_i \quad (2.2)$$

with the joint configuration θ of the robot, and the singular values $\sigma_1, \sigma_2 \dots \sigma_n$ of the Jacobian J [YOSH85b, p.3–4].

The **condition number**¹¹ quantifies the transformation errors between the Cartesian and joint space. The smaller the condition number, the smaller the error propagation. At so-called

¹⁰Further insight into the metric is provided in the appendix D.

¹¹Further insight into the metric is provided in the appendix D.

"isotropic points" ($\kappa = 1$), no error propagation occurs, which is valid for an orthogonal Jacobian with vectors of equal magnitude [SAL82, p.9] The condition number is computed as indicated in equation 2.3.

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}} \quad (2.3)$$

with σ_{max} and σ_{min} as maximum and minimum singular values of the Jacobian. For a homogeneous Jacobian, the calculation can be simplified, resulting in equation 2.4:

$$\kappa = f(J^T(\theta)) = \|J^T\| \|J^{-T}\| \quad (2.4)$$

If the Jacobian is not of full rank¹², $\sigma_{min} = 0$ follows and the condition number becomes infinite $\kappa \rightarrow \infty$ [TANE00, p.69].

The **dexterity index**¹³ D calculates the percentage of reachable orientations for a position in the robot workspace and is computed in equation 2.5.

$$D = \frac{1}{3} * \left(\frac{\delta_\gamma}{2 * \pi} + \frac{\delta_\beta}{2 * \pi} + \frac{\delta_\alpha}{2 * \pi} \right) \quad (2.5)$$

with the reachable range of the rotation angles δ_i for the respective angles roll α , pitch β and yaw γ in relation to the respective full angle range of $2 * \pi$ [TANE00, p.65]. The numerical value "reachability index" D_{RI} (see equation 2.6) was proposed to provide a numerical calculation approach.

$$D_{RI} = \frac{F}{M} \cdot 100 \quad (2.6)$$

D_{RI} is calculated per position as the percentage of reachable orientations F over the total number of tested orientations M . Both D and D_{RI} are local, kinematic, intrinsic, and relative metrics and represent the range or amount of reachable orientations per position.

Table D.1 provides a non-comprehensive list of metrics and classifies them according to the aforementioned criteria. An excerpt of the table is given below (see table 2.1), with range as local (L) or global (G), physical reference as kinematic (K) or dynamic (D), the task reference as intrinsic (I) or extrinsic (E), and the metrics scale as absolute (A) or relative (R). The upper part of the table contains global metrics taken from robot data sheets. These are typically used to select a robot for a task or family of tasks when purchasing the robot or allocating the robot to an assembly station (as applied in chapter 4). While these global metrics are well-defined and standardised among robot manufacturers [RUSS22, p.6], they cannot ensure executability without consideration of additional local metrics. The lower part of the table classifies local metrics utilised to evaluate the pose-dependent behaviour of the robot. Some local metrics will be applied for workspace evaluation and base placement in chapters 5 and 6.

¹²If the rank of the Jacobian is lower than the dimension of the task m (e.g. three-dimensional or six-dimensional etc.) $rank J(\theta) < m$, a *singularity* of the robot occurs. Thus, when the determinant of the Jacobian approaches zero, the robot approaches a singularity [TANE00, p.6].

¹³Further insight into the metric is provided in the appendix D.

Table 2.1: Excerpt of table D.1: Overview and classification of robot performance metrics.

Metric	Explanation and source	Range	Physics	Reference	Scale
Reach	[m]	G	K	I	A
Payload or maximum load	[kg]	G	D	I	A
Condition number κ	Kinematic isotropy of the Jacobian [GOSS91, p.113; TANE00, p.69]	L	K	I	R (0, ∞]
Manipulability measure w	Distance to singular values; Measures "how easy" it is for the robot to move out of the current pose. (see Eq. D.2) [YOSH85b]	L	K	I	R
Dynamic manipulability measure	Extension of the manipulability measure to reflect the dynamic properties [YOSH85a]	L	D	I	R
Joint velocity measure	Compares current and needed joint angle	L	K	E	R
Dexterity index D	"a measure of a manipulator to achieve different orientations for each point within the workspace" [TANE00, p.65]	L	K	I	R
Workspace evaluation measure G	Combines the percentage of free workspace area (not occupied by objects), the percentage of reachable objects in the workspace and the average manipulability [WOJT20, p.188].	G	K	E	R [0,1]
...	...				

2.2.3 Workspace representation with combined metrics

Performance metrics are utilised to design, compare or place robots. As often not a singular metric provides all necessary information about the robot's capabilities, approaches investigate the combination of metrics and optimisation thereof [WOJT20]. Below examples of approaches with several metrics are given.

Early works of KLEIN ET AL. (1987) consider the determinant of the Jacobian, the condition number, the minimum singular value, and the joint range availability (see appendix D for details on the metrics) as the four metrics determining the dexterity [KLEI87, p.73].

The **capability map** was invented by ZACHARIAS ET AL. (2007) as a means of describing the reachability of a robot in a visual and human-interpretable way. To generate the capability map, a discrete number of orientations are tested for reachability by the TCP at discrete positions to approximate the reachability and, thus, the robot workspace. Therefore, the robot's reachable workspace is discretised in equidistant voxels (cubes). In each voxel, the centre is chosen as position (x,y,z). Around this position, a sphere is computed. Vectors between equidistant points on the sphere and the centre position are taken as robot flange orientations to calculate reachability for each pose. The percentage of reachable poses per voxel over tested poses is defined as the reachability index D_{RI} . Figure 2.7 visualises a cross-sectional view of a so-called

capability map of a UR10 robot arm. The reachability index indicates the percentages of reachable orientations at discrete positions in the robot's workspace.

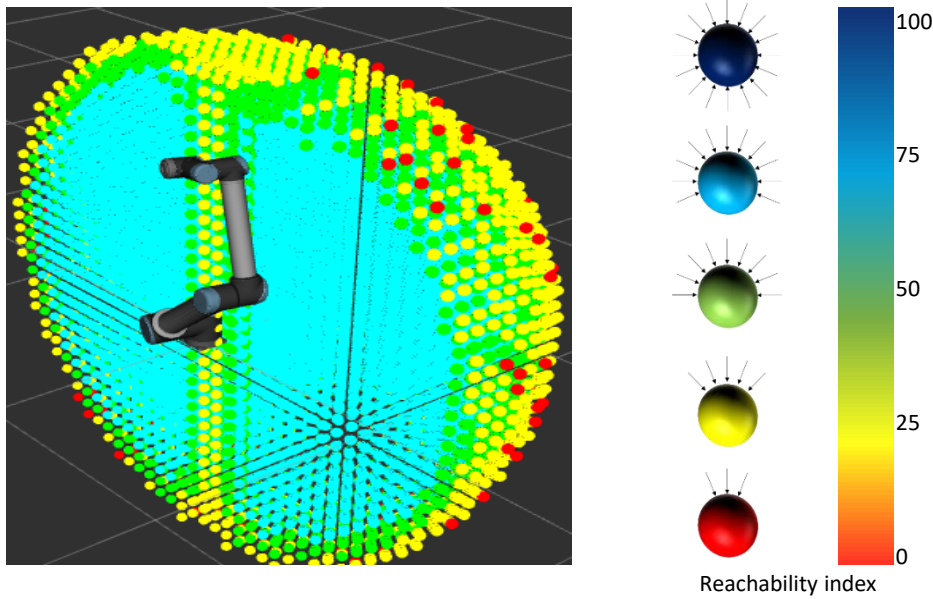


Figure 2.7: Cross-sectional view of a capability map of a UR10 robot arm, in accordance to [ZACH13; ZACH12], visualised in ROS with the library Reuleaux by [MAKH18]. Thereby, the colour range indicates the reachability index — the number of reachable orientations per tested position. [ZACH13; ZACH12]

The method of generating a capability map was transferred into Reuleaux — a library for ROS by MAKHAL ET AL. (2018). They claim that the Reachability Map is not sufficient to represent the reachability of robots but follow the same steps in creating their reachability representation, as can be derived by comparing the pseudo-code in [MAKH18, p.2] and pseudo-code in [ZACH13, p.11]).

A similar approach to represent the workspace was presented by CASTELLI ET AL. (2008), who analogously discretised the workspace into cubes and calculated how often a configuration was found to reach the position in the centre of the cube. In difference to ZACHARIAS ET AL., their solution was represented as a matrix with values indicating if and how often a solution for each cube was found, thus not choosing a visual and human-readable solution.

VAHRENKAMP ET AL. (2012) extend the capability map with discrete values of the manipulability measure for all voxels. In later works VAHRENKAMP ET AL. (2015) present the **IK-map** as an extension of reachability and capability maps by storing specific Inverse Kinematics (IK) solutions to later seed iterative IK-solvers. In addition to the reachability value, the extended manipulability is stored. [VAHR15]

ZHANG, CURSI, AND YANG (2019) present "WSRender" as an open-source Matlab-based tool to combine different evaluation indices to evaluate the workspace of robots in varying environments. Metrics¹⁴ can be chosen and computed for equidistantly discretised cubes and stored in a weighted workspace map. [ZHAN19b]

¹⁴Explicitly: the kinematic and dynamic manipulability measure, condition number, minimum singular value, order-independent manipulability, harmonic mean manipulability measure, and isotropic index. Appendix D details the metrics.

WOJTYNEK ET AL. (2020) propose computationally aided placement of workpieces in a robot's workspace, but optimising the workspace evaluation measure G . G is a weighted function of area, reachability, and manipulability. [WOJT20] Since the metrics are evaluated equally, unreachable poses may be preferred. Optimising robot design or base placement for multiple metrics often results in a Pareto front of the mixed integer optimisation problem [TRAB22, p.8].

Currently, no workspace representation supports choosing suitable metrics to evaluate the executability of specific assembly tasks. The integration into robot simulation and control is limited¹⁵ All workspace representations have been applied for base placement, which will be discussed in the following section.

2.2.4 Robot base placement

Placing mobile robot bases aims to find base placements that allow for the executability of all tasks by the mobile robots. The base placement relies on quantitative metrics, assessing the executability of assembly tasks in the robot's workspace, as presented above. The **valid workspace** is defined as the area in the robot workspace wherein all task poses are reachable, the robot is unobstructed by obstacles and itself, and the paths between task poses are free from collisions, making them appropriate for the intended application [LUET92, p.1105]. Placing the robot so that all assigned tasks lay in its valid workspace is the precondition for all tasks. Thus, evaluating the areas in the robot's workspace that provide executability is a precondition for robot base placement [VAHR13; WEIN21; MAKH18].

Automatically placing robots involves iterative calculations of IK solutions, utilising search or optimisation algorithms, or inverting a discretised workspace evaluation (commonly known as "maps") to solve the problem [VAHR13; MAKH18]. Iterative brute force exploration of the solution space, as well as multi-criteria optimisation in combination with IK solvers, can result in lengthy computation times and solutions close to singularities [PELL17, p.103]. On the other hand, achieving high-resolution reachability data in, e.g. a capability map requires a large storage memory, limiting the use of these methods. [MAKH18; BURG15; ZACH09; ZACH08]

As can be found from analysing the literature, base placement problems can be categorised regarding the problem formulation and the solving method. Problem formulations differ regarding the representation of tasks — coverage tasks as in [HASS16; HASS18] or singular tasks as in [VAHR13] — regarding the distribution of tasks — reachable from a single base placement (e.g., [SON19]) or several base placements (e.g., [KALA18]) — representation of task executability — predefined distance is assumed to be reachable (e.g., [FORS20]), reachable task poses (e.g., [MAKH18; VAHR13]), several performance metrics have to be satisfied (e.g., [WOJT20]). HASSAN ET AL. (2016) propose a mathematical model and solve it for an optimised sequence of base placements for a team of robots, whereby calculating the uncertainty in the exact base placement of the robots, resulting from real-world uncertainties like the uncertainty of measurements as input for positioning and navigation. Whilst measurement uncertainty is accounted for, this approach could be extended to include uncertainties resulting

¹⁵Reauleaux provides a library for ROS[MAKH18]. Other applications rely on Matlab.

from the precision of robot movement, e.g. a specific position cannot be configured more precisely in dependence on the robot's topology. FORSTENHAUSLER ET AL.(2020) formulate an optimisation problem, minimising the sum of the distance of the robot goal poses and the centre of the workspace. If not all points can be reached by one base placement, the points are randomly divided into two groups until all points can be reached from one pose. Apart from distance, the reachability and executability of the tasks are not validated. Additional approaches for addressing this problem have also considered minimising energy consumption, actuator torques, and forces in the context of an orthoglide parallel robot [UR-R10]. In MITSU ET AL. (2008), base placement is combined with TCP path planning and solved by a combination of genetic algorithms, quasi-Newton optimisation, and a constraint handling technique.

Existing methods of robot base placement primarily concentrate on positioning robots for coverage tasks, often involving uniform robots and homogeneous task types, rather than distributed discrete tasks, heterogeneous task types and heterogeneous robot types. For applying base placement in assembly scenarios with a wide variety of products, the need arises to expand these methods to accommodate diverse tasks and various robot types.

Found base placements in combination with allocated tasks to the robot at the base placement as the planned formation form the input for the subsequent execution stage of process execution, as illustrated in 2.1. For collision free and successful process execution localisation, navigation, path planning, motion planning and control have to be carried out. [SHI19; CHEN22; ZHAN23; XU20]

2.3 Semantic description of assembly robot capabilities

The forthcoming section provides an overview of state-of-the-art description models within the assembly domain, with a primary emphasis on the depiction of robots and their associated capabilities. At first, an overview of modelling approaches is given in section 2.3.1, focusing on ontologies, as they are flexibly adapted to new assembly scenarios [FROS22, p.6]. Secondly, description models of capabilities are investigated in section 2.3.2. Section 4.2.2 compares previous description models regarding their applicability here. Section 2.3 lies the knowledge foundation built upon in chapter 4.

The foundation of assembly automation is preparing implicitly existing expert knowledge in explicit form by describing it in a formal, machine-readable format and thereby rendering it accessible [BRAN20, p.32–33]. In smart manufacturing, the degree of interoperability of hardware and software components depends on the quality and flexibility of their information models. The prerequisite for intelligent, artificial systems—from autonomous robots to decision-support systems—is a technical representation and processability of knowledge. [BEIE19, p.1; HATV78, p.896] Consistent modelling based on a digital representation of the physical objects of production systems is necessary for smart manufacturing [GÖPP21, p.3]. Knowledge bases are central to an agent's reasoned action, robustness, flexibility and autonomy, as it makes decisions based on current knowledge [BEIE19, p.5]. Drawing conclusions from existing knowledge is a central aspect of intelligent behaviour [BEIE19, p.20]. Along these lines, the

term "intelligent" refers to the ability of a system to infer¹⁶ the implicit consequences of its explicitly presented knowledge [BAAD03, p.3]. The goal of modelling assembly tasks, robots, and their capabilities is a formal semantics that allocates an unambiguous meaning to logical statements.

According to BRANDMEIER (2020), knowledge can be regarded as the most important resource of companies [BRAN20, p.7]. The availability of knowledge is a basic prerequisite for providing services in the form of products or services. Therefore, the design of the knowledge base as the totality of the company's available knowledge is of great importance. [AMEL13, p.1] Therefore, different architectures and knowledge bases aiming to structure and standardise the vocabulary in the manufacturing domain evolved.

One of the most commonly used approaches for representing components in the manufacturing domain was first proposed by STEINWASSER (1997). Their model represents information, processes and functional components within product and process planning abstractly by distinguishing between products, processes and resources, known as PPR. Following the design, VAN BRUSSEL ET AL. (1998) propose the Product, Resource, Order, Staff Architecture (PROSA) as a reference architecture for holonic manufacturing systems, comprised of three core holons: product, order, and resource. Various architectures have been derived from PROSA, distinguished by architecture, scope, paradigm, distribution, order, and pattern. Figure 2.8 contains the three partial models of PPR and their relations. In the first partial model, **product**, component types are initially created, and their corresponding component variants are stored. In the second partial model, **process**, assembly operations and their sequence are represented as a precedence graph. The combination of the product and process models constitutes the requirement side. In contrast, there is the third part, the **resource** partial model. When all information carriers are interconnected, a closed loop is formed, highlighted by arrows. [MÜLL15, p.555]

2.3.1 Ontologies in the manufacturing domain

Ontologies have been used in the manufacturing domain to simplify robot and data integration and transfer [ROBO15, p.viii], derive code for robot programming and incorporate explicit knowledge from sensory feedback [OLIV19], as well as capability-based task allocation in Multi-Robot Systems (MRS) [KLUG22b].

The goal of utilising ontologies is the declarative representation of the knowledge base of a domain. Everything that "exists" can be represented in knowledge-based systems [GRUB93, p.199]. Ontologies represent semantic relationships between information according to syntactic, i.e. machine-readable, rules. Ontologies and semantic web solutions enhance the interoperability in manufacturing [MONO15, p.2].

Definition: One of the most cited definitions of ontology in the literature is Gruber's, which defines ontologies as an explicit specification of a concept through a formal description of a shared vocabulary for a domain and the relationships between concepts formed from that

¹⁶Inference is deducing new knowledge B from given knowledge W , whereby the relation R between the two is referred to as the inference relation. [BEIE19, p.20]

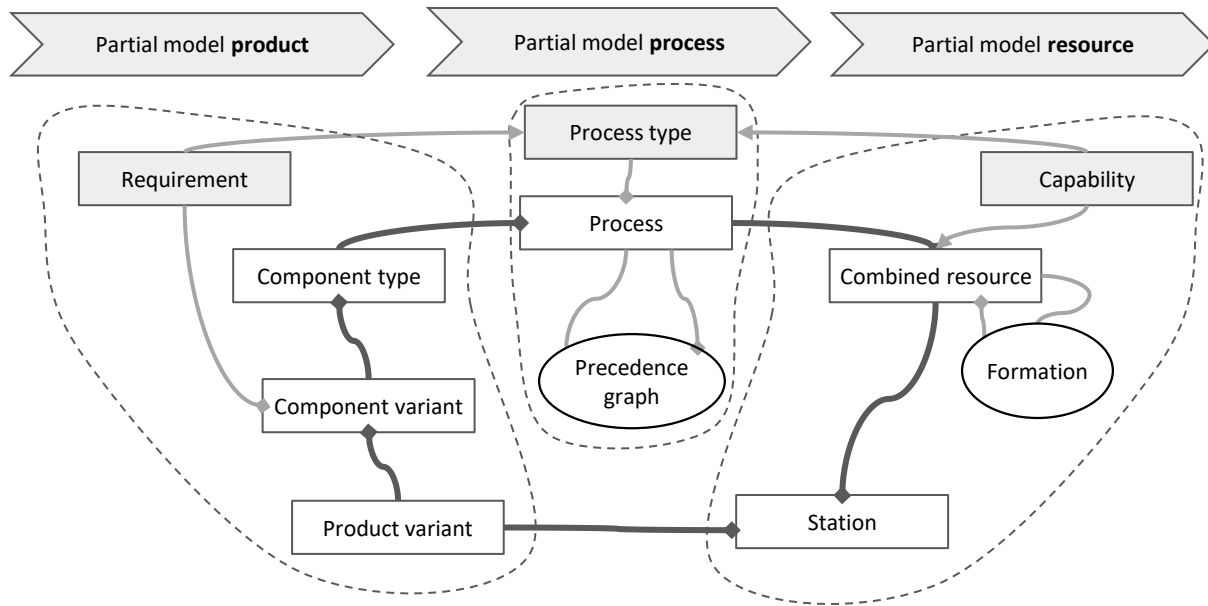


Figure 2.8: Class diagram of the Product, Process and Resource (PPR) model, representing components in the manufacturing domain, adapted from [MÜLL15, p.555].

vocabulary [GRUB93, p.199]. Ontologies comprise the formalised and explicit specifications of concepts and relationships (both taxonomic and semantic) that exist between concepts [STUD98, p.184; LEMA06]. Ontologies enable a common understanding of a domain and facilitate communication between people and technical systems. Therefore, ontologies comprise both the conceptual description of a domain (semantics) and the specification of this conceptualisation in the form of a formal description (syntax) [GRUB93]). A formal definition of ontologies was only introduced in 2009. According to GUARINO ET AL. (2009) a conceptualisation in mathematical interpretation is a relational structure [GUAR09]

Ontologies model knowledge in the form of „individuals, classes, functions, relations, and axioms.“¹⁷ [OLIV19, p.3; GÓME04, p.253]. *Individuals* represent the things, the objects, which are conceptualised in the ontology. *Classes* summarise these individuals depending on their properties, representing a set of individuals. Classes can contain multiple classes, resulting in an interconnected network of individuals defined by their properties, which are either asserted directly or inherited indirectly through their hierarchy. *Functions* relate the individuals to each other and offer the possibility to identify them. *Relations* connect individuals. *Axioms* are expressions in the language that use the previous elements to determine what is true in the ontology, e.g. relationships between entities. [OLIV19, p.3] Axioms convey assertions regarding relationships among entities within the relevant domain. The *vocabulary* includes all specific domain-relevant objects and their relations. [BEIE19, p.179]

In recent years, there has been a substantial influx of ontology proposals within the domain of changeable production and assembly. Existing ontologies built for and applied in the manufacturing domain agree on modelling the three instances of PPR as introduced before (see figure 2.8). The applications of ontologies in assembly are manifold and pursue changing goals:

¹⁷Concepts and classes can be used synonymous, as well as individuals and instances.

- Overarching goal: knowledge representation, reasoning and a standardised understanding of concepts [ROBO15, p.viii]
- Interoperability through simplified data integration and transfer [ROBO15, p.2]
- Interoperability through simplified robot integration and substitution [ROBO15, p.2; WESE20, p.1685]
- Code creation for robot programming [OLIV19, p.2; TRAB22]
- Incorporation of explicit knowledge from sensory feedback, rendering human interaction for knowledge generation unnecessary [OLIV19]
- Task allocation based on capabilities [JÄRV19]
- Automated process planning [WESE20, p.1679]

Reducing the manual programming effort for ever-changing flexible assembly stations, it is desirable to utilise knowledge bases to deduct robot code from high-level action descriptions [KUNZ17]. Robot programs are currently written for specific robots and applications in specific environments. For future reusability of code and programs, the goal is to write robot programs independently of the robot model, task and environment and to define them in separate knowledge bases [OLIV19, p.2]. BESSLER ET AL. (2018) propose an ontology-based assembly sequence planner, deducing the next action based on the currently available knowledge about the product, the process steps and the environment. The ISO norm 15531 defines MANDATE, a PPR-based standard for interoperable data exchanges in the manufacturing domain. The differentiate manufacturing capability models and product models [CUTT07, p.5]. MABKHOT ET AL. (2020) propose an ontology-based decision support system focused on finding formations of robots based on the current formation and the planned processes in RMS. Therefore, the assembly system is modelled in "modules", which express certain capabilities with certain parameters. Compared to the application here, capabilities refer to the quality rate and amount of manufactured parts. The spatial capabilities and, explicitly, robots are not considered. LEMAIGNAN ET AL. (2006) introduced a semantic network called MAnufacturing's Semantics Ontology (MASON) as the upper-level ontology for manufacturing, encompassing the product, process and resources modelling approach. However, it does not include a representation of capabilities.

One of the first approaches to modelling robots and robot control is presented in KnowRob, which focuses on the interaction between a robot and its environment and modelling actions. The open-source knowledge processing system was introduced in (2009) by TENORTH ET AL. and is continuously updated [TENO13; BEET18]. KnowRob is built upon the upper ontology Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) + DnS Ultralite (DUL) [TENO13]. Its second generation focuses on rendering, simulation and integration of the modelling into the control system [TENO13, p.514].

2.3.2 Capability models

Formation planning of multipurpose and combined robots necessitates representing the robots' inherent capabilities to allocate tasks and spatially place them in changeable assembly stations.

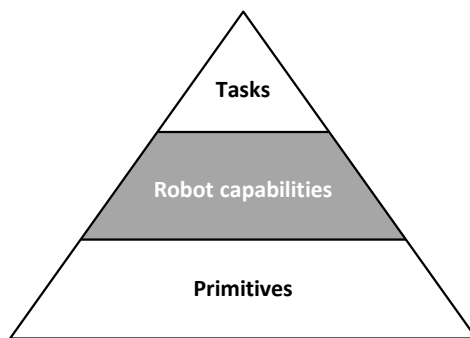
Previous research proposed different semantic representations of robot capabilities. In the following, capabilities will be defined and an overview of capability models is given.

Two central goals for modelling capabilities can be identified when analysing research: deriving robot code and matching the requirements of the tasks to the available capabilities of the robots at hand. [OLIV19; KLUG11] By extending the PPR model with capabilities, firstly, the relationship between product, process, and resource is formally defined. Secondly, the fixed character of linking individual resources to specific processes needed to produce specific products can be dissolved by linking in dependence on required and offered capabilities [PFRO13; ONOR06, p.327; KLUG11, p.77]. The available capabilities create a flexibility corridor in which tasks can be allocated to resources by matching the requested and available capabilities [MÜLL15, p.554–555; FROS22, p.5]. Robots gain knowledge about themselves by linking descriptions of robot components via capabilities to actions in an ontology [KUNZ17, p.1, 4].

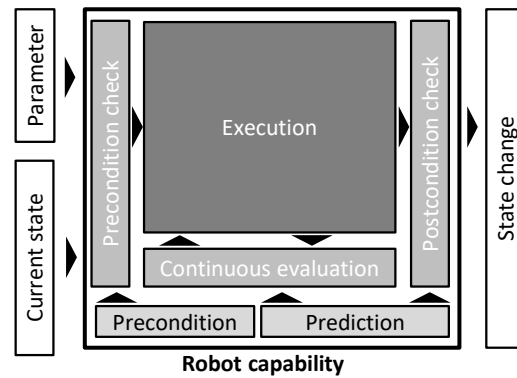
The consensus on differentiating capabilities into composed and elemental types, as observed across [PEDE16; WESE20; VERE90; JÄRV19]), is not matched by a unified understanding [GINS22, p.3; FROS22, p.3; NORM16, p.26; PERZ19, p.2]. **A capability semantically abstracts the functionality of a subject to cause an effect on an object in a specific context.** The effect can be described by the change in property values of the object, i.e. the product. The subject, also called a resource, exhibiting the ability to cause the effect can be described by corresponding properties. Capabilities are often defined in conjunction with the necessary hardware and software. [OLIV19, p.8; WESE20, p.1680] In some contexts, the abstract modelling of a "capability" and its hardware-dependent implementation, the "skill" as an executable function of the capability, is differentiated [WESE20, p.1681; VIEI23, p.144; PERZ19, p.1]. Moreover, the compatibility of robots (e.g. the hardware and software interfaces of a robot arm and a gripper) is modelled as non-value-adding skills. [GINS22, p.9] ONORI ET AL. (2006) highlight the necessity to differentiate between higher-level and basic capabilities and the combinatorics thereof. According to KLUGE ET AL. (2011) the manufacturing process standards by Verein Deutscher Ingenieure e.V. (VDI) constitute the foundation of the definition of capabilities [KLUG11, p.78].

PEDERSEN ET AL. (2016) propose capabilities (therein: "skills") to ease the process of reprogramming robots, enabling non-experts. **Capabilities are abstract representations, decoupling the tasks and programming from the hardware.** [PEDE16, p.6] Modelling capabilities enable implementation-independent planning of the capabilities [VIEI23, p.141]. Task-level programming separates an action into a series of tasks or task primitives. BØGH ET AL. (2012) differentiate between motion primitives, capabilities and tasks, whereby they successively makeup one another, as depicted in figure 2.9 a). Thereby, entire robot programs or tasks can be created from combining primitives to capabilities to tasks [PEDE16, p.4]. The robot can interpret and execute those primitives without providing specific poses for the robot [PEDE16, p.3].

Depending on the state and starting parameters, fulfilment of the preconditions for the execution are controlled, the capability is executed, the execution is confirmed, and the state is set to finish as conceptualised in figure 2.9 b). BØGH ET AL. (2012) identify 13 capabilities from Standard Operating Procedures (SOPs) in logistics and as assistive tasks: *move to*, *locate*,



a) Hierarchic levels of representing tasks as capabilities as primitives. By combining primitives, capabilities are formed. By combining capabilities tasks are formed.



b) Conceptual model of a robot capability. The fulfillment of preconditions for execution, execution of capabilities, confirmation of execution, and setting the state to finish, depend on the current system state and initial parameters and are continuously evaluated.

Figure 2.9: Modelling robot capabilities and relating them to primitives and tasks, in accordance to [PEDE16, p.5–6; BØGH12].

pick up, place, unload, shovel, check, align, open, close, press, release, turn. Later, the authors extend the concepts to include services as planning layers to orchestrate the provided primitives and capabilities [SCHO18, p.73].

2.3.3 Ontology development methodologies

The process of ontology development necessitates a systematic approach during development. Therefore, established methodologies are considered in the following.

The proposed **"Pipeline for ontology-based modelling and automated deployment of digital twins for planning and control of manufacturing systems"** [GÖPP21] shown in figure 2.10 defines three steps to integrate digital twins into cyber-physical systems. The digital twin is a fully connected digital model linking physical and digital assets. 1) The initial phase of the pipeline considers the definition of a schema for modelling the domain and application of interest. Based on a manual selection of established meta-models and ontologies and partial or full application of the provided modelling structures, a domain ontology is created. 2) The subsequent phase involves standardised and structured modelling, incorporating static data specific to use cases. 3) The ultimate phase of the pipeline involves deploying digital twins into operation, ensuring a complete connection between the digital model and the broader cyber-physical system. Even though the central intent of the pipeline is the creation of a digital twin, the first two steps aim at modelling and can, therefore, be applied here.

SURE ET AL. (2009) structure the development and set-up process of the ontology into five steps, with a specific emphasis on the corporate perspective. "The phases are "Feasibility Study", "Kickoff", "Refinement", "Evaluation" and "Application and Evolution". [SURE09, p.138] 1) The feasibility study identifies the need for an ontology and its technical feasibility. 2) During the kickoff, the requirements to be answered in the ontology are gathered, existing knowledge sources are identified, and the reuse of existing ontologies is considered. 3) During refinement, the formal ontology is generated, and the relation between concepts is implemented.

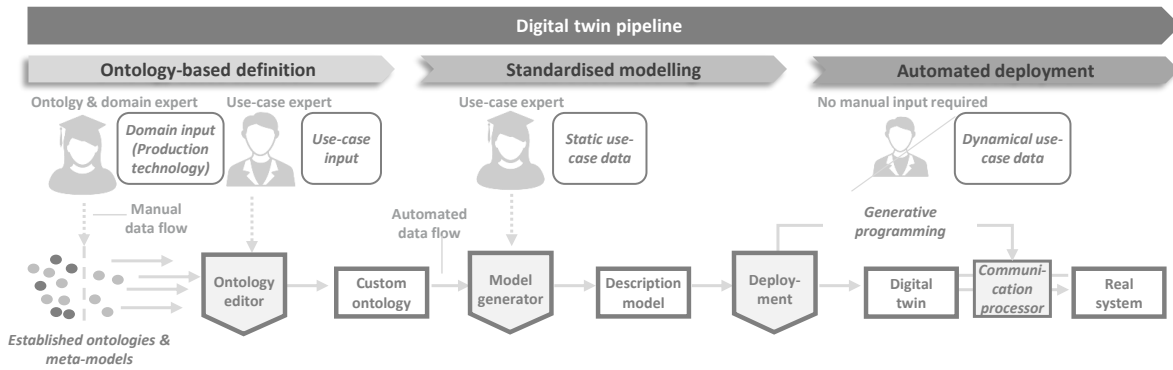


Figure 2.10: Ontology development as part of the digital twin pipeline, based on [GÖPP21].

4) Ontologies can be evaluated formally, regarding user satisfaction and technologically (e.g. scalability or interoperability). 5) Finally, the utilisation phase in operation follows.

In difference, NOY AND MCGUINNESS (2001) focus solely on the development of the ontology itself, thereby they differentiate the second and third steps of SURE ET AL. (2009) into more granular seven steps. These steps are 1) the determination of the domain and scope of the ontology, 2) the consideration of reusing existing ontologies, 3) the enumeration of essential terms in the ontology¹⁸, 4) the definition of the classes and the class hierarchy, 5) the definition of the properties of classes, 6) the definition of the facets of the properties and finally, 7) the creation of instances.

2.4 Task clustering

The following section provides an overview of clustering algorithms, as clustering algorithms will be applied to group distributed task poses to assign them to robots at specific base placements in chapter 6. Therefore, the clustering algorithms presented here are chosen for the intended use.

Clustering is a procedure for discovering similarity structures in (usually relatively large) data sets [ANKE99, p.49]. Groups or subclasses of "similar" objects found in this way are called **clusters**, and the assignment into groups is called **clustering**. Task clustering has been studied primarily in the context of search and rescue operations and scheduling and sequencing problems [JANA17; EZUG22, p.32].

Clustering algorithms can be classified into partitional and hierarchical clustering, as visualised in figure 2.11. Hierarchical clustering algorithms organise data objects into hierarchical levels. Through either a top-down or bottom-up process, clusters are progressively created to construct a dendrogram (structured like a tree) representing the hierarchical arrangement of the formed clusters. In difference, partitional clustering algorithms cluster data into a nested sequence of groups devoid of any hierarchical arrangement. Partitional algorithms systematically divide the data set into a fixed number of distinct subsets, K , through an iterative process that optimises a specific criterion function. These density-based algorithms seek to identify regions of high

¹⁸As step three has the characters of making notes, which will be detailed in steps four and five, step three will not be detailed here.

density in the data space that are separated by areas of low-density noise. [SISO12, p.83; EZUG22, p.4–11]

Partitional clustering can be further divided into algorithms that allow for the multi-assignment of data points to clusters (mixture and fuzzy clustering) and hard/crisp clustering, where each data point is assigned to a single cluster. Graph-theoretic clustering represents clusters within graphs, where a graph comprises nodes connected by edges. Nodes are clustered so that inter-cluster edge density is lower than intra-cluster edge density. In model-based clustering, it is assumed that data originates from an underlying probability distribution or model. Density-based and square error clustering provide clusters based on the spatial closeness of the data points and tend to derive spherical clusters.

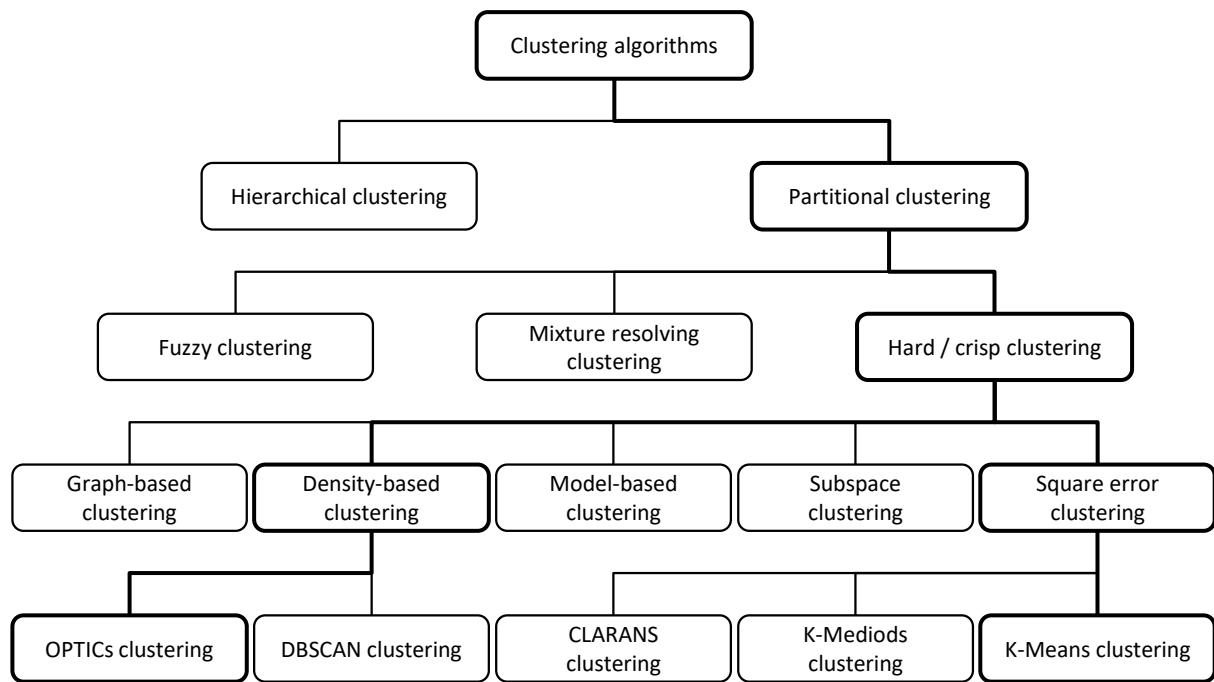


Figure 2.11: Classification scheme for clustering algorithms, based on [EZUG22].

The partitional clustering algorithm **K-Means** is categorised into the class of square error clustering methods. The K-Means algorithm is a widely used vector quantisation and cluster analysis method. [SISO12, p.83] K-Means partitions a data set $d_1, \dots, d_l$ consisting of L observations of a variable d into K clusters, where K is a predefined number. Each cluster consists of data points whose inter-point distances are small compared to the distances to points outside the cluster.

K-Means aims to divide the data set into K clusters so that the sum of squared deviations from the cluster centroids is minimal. [EZUG22, p.9] K-Means assigns each data point d_l to the nearest (by Euclidean distance) cluster centroid μ_k with $k = 1, \dots, K$. The objective function A to be minimised is given in equation 2.7 [BISH06, p.444–445].

$$A = \sum_{n=1}^N \sum_{k=1}^K r_{l,k} \|p_l - \mu_k\|^2 \quad (2.7)$$

Where $r_{l,k} \in 0, 1$ is a binary indicator variable describing which of the K clusters the data point d_l is assigned to. (If the data point d_l is assigned to cluster k , then $r_{l,k} = 1$ and $r_{l,j} = 0$ where $j \neq k$.) K-Means aims to find the values for $r_{l,k}$ and μ_k that minimise B . The iterative solving mechanism is structured as follows:

Algorithm 1 K-Means clustering algorithm.

```

Select  $K$  arbitrary centroids  $\mu_k$  as initial centroids
repeat
    Assign each data point  $d_l$  to the nearest centroid  $\mu_k$ 
    Adjust the centroids to the means of all data points assigned to them
until Centroids converge

```

K-Means is widely used since it can rapidly determine the cluster centroids. The algorithm prioritises groups with low variance and comparable sizes. K-Means clustering can only guarantee local and not global convergence since the choice of initial cluster centroids significantly impacts the algorithm's outcome. Another drawback is that the number of clusters needs to be specified before executing the algorithm. Outliers and noise impact the identified clusters. K-Means is limited to identifying clusters of a circular shape. [EZUG22, p.10]

As an extension of K-Means, **K-Medoids** or Partitioning Around Medoids (PAM) uses the medoid (the most centrally located pattern in the cluster) of each cluster to represent the cluster, which makes it more resilient to outliers than K-means. Besides using medoids instead of centroids, its working principle is analogue to K-Means.

The **Clustering Large Applications based on Randomized Search (CLARANS)** algorithm is a modification of the K-Medoids algorithm that uses random input data samples to determine the best medoids instead of processing the entire data set. This makes it more effective than K-Medoids when dealing with dense data sets. However, the algorithm may produce incorrect clustering results if one or more sampled medoids significantly differ from the best medoids. [SCHU18]

The **Density-Based Spatial Clustering of Applications with Noise (DBSCAN)** algorithm is a non-parametric method for identifying high-density regions in a data set and distinguishing them from low-density regions. It also detects clusters of data points and noise points that do not belong to any clusters. The algorithm identifies core samples, which are data points located in high-density areas. [ESTE96, p.4] DBSCAN can identify clusters with arbitrary shapes, is robust to outliers and data input order, and has a $O(N^2)$ complexity. [SISO12, p.86] DBSCAN provides different outputs depending on input parameters and settings.

Ordering Points to Identify the Clustering Structure (OPTICS) is an algorithm that detects density-based clusters in spatially distributed data. According to its founders ANKERST ET AL., OPTICS improves DBSCAN, which struggles to identify meaningful clusters in data with varying densities, by ordering the points in the database linearly so that the closest points become neighbours [ANKE99, p.52]. Each point is assigned a distance representing the required density for a cluster so that both points can be included. This information is organised in the form of a dendrogram [PEDR11].

Unlike DBSCAN, OPTICS processes data points in a specific order to produce consistent results. The algorithm works by considering an infinite number of densities that is less than a "generating distance", similar to DBSCAN. However, instead of assigning clusters to data points, OPTICS records the order of data point processing. An extended DBSCAN algorithm can then use this ordering to assign data points to clusters.

2.5 Verification and validation methodology

According to the DSR research methodology, the conducted research result in research artifacts. These artifacts must be verified and validated in their application field, fulfilling the relevance cycle. The key questions to be investigated in verification and validation are: *Do the designed artifacts answer the initially derived research demand? How can an improvement in the research subject be measured?* [HEVN07, p.89] The central goal is a systematic analysis and evaluation of the artifacts, thereby evaluating the degree of usefulness of the developed artifacts. [NATI75, p.27] The validation test is designed to ensure that the model accurately reflects the behaviour of the actual system. [RABE08, p.15] The aim of the conducted verification and validation is to provide a systematic basis for an acceptance decision of the entire framework of formation planning and to document it comprehensibly. The goal of empirical testing of the created models is to capture typical problems, validate the developed artifacts in the context of their practical application, and investigate the application context itself. [ULRI81, p.7]

According to SARGENT (2010), **model validation** is the process of substantiating that a model provides a satisfactory and acceptable range of accuracy within its application domain and intended purpose. OSMAN BALCI (1997) states: "Model validation deals with building the right model" [OSMA97, p.135]. Since a model is developed concerning a specific use, validation must be performed accordingly. A model may be valid for a particular set of experimental conditions yet invalid for another. To validate a model, its behavioural accuracy for the intended application domain has to be measured [OSMA97, p.135].

Model verification ensures building the model correctly [OSMA97, p.135]. Thus, the correctness of the programming of the source code and implementation of the conceptual model has to be tested [SARG10].

SARGENT (2010) structures the process of model generation as follows: First, a conceptual model is developed, followed by conceptual model validation. This process is repeated until the conceptual model is satisfactory. Secondly, the computerised model is developed based on the conceptual model followed by computerised model verification. This process is repeated until the computerised model is satisfactory. Last, operational validation is conducted on the computerised model. To validate the simulation model, SARGENT (2010) defines the following four essential steps to validate and verify a simulation model [SARG10, p.168]:

- The **conceptual model validation** determines the correctness of underlying theories and assumptions of the conceptual model and that the model reasonably represents the problem entity in dependence on the model's purpose.

- The **computerised model verification** assures the correctness of the implementation and programming of the model.
- The **operational validation** determines sufficient accuracy of the model's output behaviour in dependence on the model's purpose for the entire range of the application domain.
- The **data validity** ensures the correctness and applicability of the data, which is needed for building, evaluating and testing the model.

Throughout the thesis, the structure of dividing the verification and validation into four distinct steps as proposed by SARGENT (2010) is followed. Thereby, different validation techniques and methods from OSMAN BALCI (1997), RABE ET AL. (2008) and SARGENT (2010) are utilised. Four categories are differentiated [OSMA97, p.138]:

- **Informal techniques** "rely heavily on human reasoning and subjectivity".
- **Static techniques** assesses the model's accuracy based on the design of the model and the source code (e.g., compiling the code).
- **Dynamic techniques** evaluate the model based on its dynamic behaviour. The first step involves the instrumentation of the executable model, while the second step entails the execution of the instrumented model. Finally, in the third step, the model output is scrutinised to evaluate the dynamic behaviour of the model.
- **Formal techniques** mathematically prove the model's correctness.

3 Formation planning of mobile robots state of the art

Upon reviewing the foundations, the lack of a systematic review exploring combinations of robot modelling, workspace evaluation, and base placement required for formation planning becomes evident. A systematic review of the state of the art is paramount for advancing beyond established concepts and ensuring rigour.

At first, the applied methodology for the literature review is provided, followed by the depiction of the chronological process. The literature review discloses the research gaps for automated formation planning. This chapter concludes by translating the research gaps into the primary and its three underlying research questions in section 3.4.

3.1 Methodological approach to the systematic literature review

A literature review was conducted to ensure rigour and identify relevant knowledge or gaps in the existing knowledge base regarding formation planning or its building blocks (namely modelling of assembly robots, workspace evaluation, base placement, and task allocation) in changeable assembly. As previous reviews solely focused on individual building blocks, conducting a comprehensive literature mapping and review that encompasses the combined scope is imperative.

A systematic approach ensures an explicit, reproducible procedure for identifying, evaluating, and synthesising the knowledge base. A predefined procedure is followed to guarantee replicability and transparency. The procedure (visualised in figure 3.1) commences with a systematic mapping of the research field [PETE08] and is concluded with a systematic review [MOHE10] of the most relevant topics. While the literature mapping structures the research area of interest, the literature review describes the found content and methods in more depth. The literature review is based on established guidelines [MOHE10; PETE08] and practices [DALI22; ANGR20; LIAO17; DOLG22; SEUR12]. The iterative nature of reviewing, as defined by SEURING AND GOLD (2012 [p.3], adapts to new research findings, with emphasis placed on presenting final mapping and review results rather than the chronological process. An overview of intermediate findings is available in appendix B.

To identify the relevant literature, the literature databases, namely Scopus [SCOP] and ResearchGate [RESE], were searched in May and June 2023. The **search strings and keywords** (see table B.1) for the review were derived from literature reviews and the foundations [PATE15; ANGE16; CHEN19b; OLIV19].

Identification: A set of 8.859 publications was initially retrieved by querying the named databases with all combinations of keywords, as detailed in the table B.1. Hereby, only the first 1000 results of Scopus and the first 500 of ResearchGate for each search string are selected to focus on eligible sources. In both databases, the appearance order is sorted by

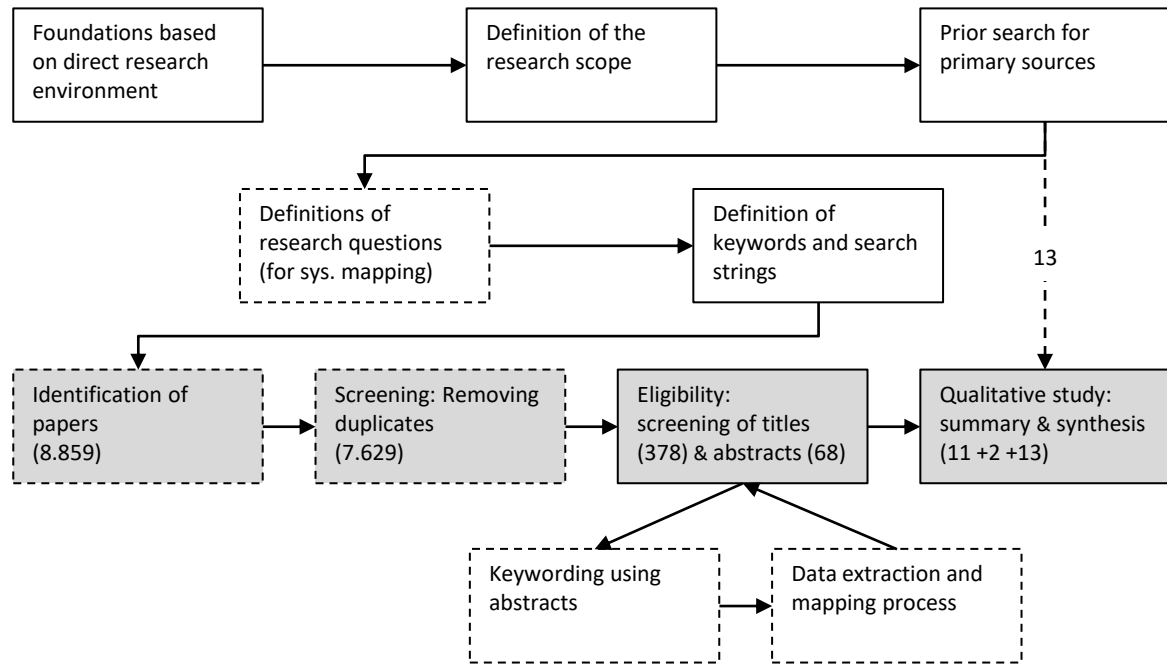


Figure 3.1: Procedure of the systematic literature mapping and review, based on [MOHE10; PETE08]. The applied literature review phases based on Preferred Reporting Items for Systematic review and Meta-Analysis (PRISMA) [MOHE10] are highlighted in grey, and literature mapping phases based on [PETE08] are framed in dotted lines.

"relevance". These relevance scores for sources are calculated based on search words' frequency, significance, position, proximity, and completeness [23b; SPAR72]. The exact calculation method of "relevance" is undisclosed. Table B.2 in appendix B gives an overview of identified sources per search string and database.

Screening and eligibility: After removing duplicates, 7.629 unique publications remained. Out of the 7.629 unique publications, 378 titles were identified as potentially relevant. Following the abstract screening, 68 were finally selected for mapping and review. Deciding on the relevance of the sources in the screening phase is based on factors such as their quality, accessibility, as well as the information provided in their title, abstract, and keywords. To ensure the reproducibility of the study, explicit inclusion and exclusion criteria were employed (as proposed in [DALI22, p.8])¹⁹.

Qualitative study: Refining the qualitative study, further criteria were applied, incorporating publication date and citation count. Consequently, sources published before 2015 were omitted, resulting in a selection of 36 sources. Additionally, the top 80 % of sources listed after the number of citations (11) were included, while sources with fewer than 19 citations (25) were excluded to ensure scientific impact. Considering the likelihood of newer publications being

¹⁹Inclusion criteria: Studies are published in peer-reviewed sources. Studies are electronically available. Studies are written in English. Studies wherein the relevance to the formation planning of mobile robots in assembly stations or its building blocks can be inferred from the title and abstract. Studies published and made available before 01.06.2023.

Exclusion criteria: Grey literature (e.g., not peer-reviewed, websites, magazines). Studies with a primary focus on scheduling, trajectory planning, navigation algorithms, sensors and their application. Reviews already discussed in section 2.1.

undercited, the sources from 2021-2023 are reviewed for enhanced relevance, adding 2 relevant sources. The notable drawbacks of the conducted review procedure are the exclusion of publications not available on ResearchGate or Scopus and sources with few citations. To overcome this, 13 additional sources were identified through an exploratory approach based on primary studies and analyses of existing reviews, resulting in a total of 26 sources for the qualitative literature study.

3.2 Qualitative literature study

The 26 sources derived from the systematic literature study undergo assessment against predefined evaluation criteria, established in alignment with the objectives of the thesis. The qualitative evaluation of the literature is summarised in table 3.2. Therefore, the upper part of the table presents the results of the systematic literature review, which is listed according to the number of citations, whilst the lower part presents the additional identified literature. The meta-data of the sources (authors and titles) is given, and the approaches therein presented are assessed regarding the fulfilment of the four categories of **Problem Formulation for task allocation (PF)**, **Base Placement (BP)**, **Workspace analysis (WS)**, and the **Applicability for Formation Planning (AFP)**. Problem formulation for task allocation refers to formulating the problem of identifying a mobile manipulator that fulfils the requirements and constraints to execute the given distributed assembly tasks. Robot base placement aims to position the mobile manipulator to ensure that all poses are placed within the "valid workspace" of the manipulator. The workspace evaluation results in a "valid workspace" that evaluates whether the task can be executed based on performance metrics indicating executability in the manipulator workspace. The criteria applicability regards the consideration of boundary conditions and interdependence between the other three criteria. The Harvey Balls indicate criteria fulfilment with the scaling given in table 3.1.

The source ZHANG ET AL. (2017) is handled separately from the general literature study, as it contains an overview of current challenges for robot cell design. Specifically, ZHANG ET AL. (2017) identify combined base placement and task allocation during formation planning, robot capability estimation, transfer of human operation to robots — on technical and semantic levels, interdependent sequence and motion planning, tool design, and human-robot collaboration as current challenges, but do not propose a solution to solve them. Thereby, the need for a deeper analysis is amplified.

The review results are detailed below. An analysis of overall compliance with the evaluation criteria, with detailed scrutiny of specific publications, is given. Finally, the presentation of transferable knowledge and identified research gaps is presented. The majority of found sources consider a combination of base placement and workspace evaluation whilst disregarding task

²⁰In addition to the initial two primary criteria, an extra 25 % fulfilment is allocated for each fulfilled criterion beyond the first two. Therefore, a different combination of fulfilled criteria can lead to the evaluation with ● and ●.

²¹The work by MALHAN ET AL. (2019) originated from the systematic literature review, but expanding, albeit less cited, studies are included.

Table 3.1: Evaluation criteria for the literature review in the field of formation planning of mobile robots in assembly.

Evaluation criteria	Fulfilment level
Problem Formulation for task allocation (PF)	○ None
	◐ Task allocation of distributed tasks considering homogeneous robots
	◑ Task allocation of distributed tasks considering heterogeneous robots
	◒ Task allocation of distributed tasks considering heterogeneous robots and their capabilities
	● Task allocation of distributed tasks considering reconfigurable robots with inheritary capabilities by the combination of resources (e.g., mobile base, end-effector, robot arm)
Base placement (BP) ²⁰	○ None
	◐ Reachability of all task poses from at least one base placement
	◑ Pose quality for task execution — required metrics for task execution are fulfilled (requires a workspace analysis)
	◒ Self-collision and obstacle collision avoidance
	● Base placement for multiple robots and distributed task poses
Workspace analysis (WS)	○ None
	◐ Inverse Kinematic is calculated, or a capability map is applied
	◑ Capability map and additional criteria are applied
	◒ Additional consideration of equipped end-effectors into workspace analysis
	● Additional matching of specific metrics to the specific tasks
Applicability for Formation Planning (AFP) ²⁰	○ None
	◐ Interdependence between two building blocks are considered
	◑ Interdependence between all three building blocks are considered
	◒ Mobile robots or manipulators are considered
	● Assembly as application field is considered

allocation. Those addressing task allocation typically disregard base placement or workspace and are therefore less represented in the literature study.

Of the found sources, only [SUEM16; PELL17; HASS18; HASC23] consider all three building blocks of task allocation, base placement and workspace analysis in some form. SUEMITSU ET AL. (2016) formulate an optimisation problem for multi-robot assembly station formations, minimising operation time and station area. They neglect factors such as the mobility of robots, the simultaneous operation of multiple robots, and the representation of objects and collisions is confined to a two-dimensional problem. PELLEGRINELLI ET AL. (2017) combine layout planning with motion planning of multiple robots in a welding cell. The method chooses the optimal solution out of all possible combinations of robots with end-effectors, base placements and collision-free trajectories of neighbouring robots [PELL17, p.100]. Due to the long computation times (~100 h) and limitation to one robot model and a maximum of 20 tasks per cell, the approach is not feasible for mobile assembly [PELL17, p.103, 114]. DAI ET AL. (2021) provide a framework for placing multiple robots for coverage tasks on large objects, based on [HASS15; HASS16; HASS18]. Their goal is to minimise the number of base placements, the makespan, and the individual torque for the execution of coverage tasks²² on large components, whilst maximising the covered surface and the individual manipulability

²²E.g. inspection, coating, spray painting.

[HASS18, p.3]. The framework consists of the following steps: 1) task pose generation, 2) base placement optimisation, 3) reachability checking for found base placements through calculating the robots' inverse kinematics, and 4) collision checking (self and environment) for found base placements. Motion planning and coverage planning are carried out for each selected base placement, and the result is evaluated regarding total coverage, makespan, torque, and manipulability. The base placement with the maximum evaluation value is chosen from the found solutions. The proposed framework exclusively addresses scenarios concerning homogeneous robots and tasks. Therefore, the formulated task allocation problem is limited to assigning areas to robots. Furthermore, the assumption that multiple task poses can be ignored and thus not executed renders the framework ill-suited for assembly processes.

HASCHKE (2023) introduces a methodology for the planning of tasks for indoor construction applications. A Multi-Travelling Salesman Problem is formulated, minimising the costs of manipulation, travel, and auxiliary processes for a mobile manipulator. The base placement of the mobile manipulator in relation to the task is assumed to be known. Therefore, no base placement optimisation is integrated. In summation, all approaches that consider all three building blocks are insufficient to be applied for the formation planning of mobile robots in changeable assembly. Thus, other sources that consider two building blocks are also investigated.

Analogously to HASSAN ET AL. (2018), KALAWOUN ET AL. (2018) propose a method to find the optimal robot base placement by discretising the space around the product. The base placements around the product are evaluated for the reachability of task poses on the product. The approach is not directly transferable to assembly as it neglects task allocation, heterogeneous robots, the executability evaluation of the task poses, and consideration of task poses for more than one product.

MALHAN ET AL. (2019) and MUTTI ET AL. (2021) investigate the reversed problem of replacing robots — placing workpieces inside a single robot's workspace. MUTTI ET AL. (2021) present an algorithm optimising the placement of workpieces in relation to collaborating robots, enhancing a performance metric over the execution trajectory. Neither task allocation nor robot mobility is considered. MALHAN ET AL. (2019) consider boundary constraints (joint limit, force and torque limits), velocity constraints, force, continuity of space trajectories configuration, collision avoidance, and singularity avoidance during workpiece placement [MALH19a, p.5588]. MALHAN ET AL. (2021) tackle the dependent problem of base placement and tool path planning, with the drawback of manually defining surface partition, job sequence, grasping points and tool paths beginning and end [MALH21, p.5–6].

LIM ET AL. (2017) optimise robotic cells for minimised operation time (approximated), layout area, and maximised manipulability by introducing multi-objective hybrid evolutionary algorithms. Sequential pairs present the cell layout. Similarly, KAWABE ET AL. (2022) employ a sequential pair representation to optimise fixed robot cell layouts by minimising the layout footprint and operation time. The approach is too slow for adaptable assembly systems due to the extensive computations required for computing the motion plan for diverse layouts and assessing the solutions.

DOAN ET AL. (2017) optimise base placement for task trajectory execution of single robots, under consideration of the constraints of singularity, reachability, joint limits and collision avoidance. FAN ET AL. (2021) place mobile manipulators for machining tasks by solving an optimisation problem, considering the stiffness, joint ranges and velocities, manipulability, and collision. XIE ET AL. (2022) propose a method for generating feasible base placements under consideration of the variable workspace of the mobile manipulator and the uncertainty of achievable base placement accuracy. The number of base placements is minimised by a greedy-based optimisation algorithm. Whilst considering distributed task poses and reachability of the task poses, no further performance metrics are considered, even though their necessity is motivated [XIE22, p.94]. ZHANG ET AL. (2020) and PAUS ET AL. (2017) utilise inverted capability maps to define base placements and plan the motion of a single robot for multiple task poses. TRABELSI ET AL. (2022) optimise the placement of multiple robot arms to be utilised cooperatively in medical applications. The placement is optimised regarding the reachability represented by the capability map, the manipulability, and the setup's compactness. Thereby, compactness is defined as the occupied volume by the robots [TRAB22, p.6].

The prevalent approach among sources to identify reachable task poses from a base placement involves either utilising a capability map [MALH19b; MALH19a; MALH21; TRAB22; ZHAN20; PAUS17] as a reachability representation or computing the IK of a robot positioned at the base placement [MUTT21; PELL17; SPEN16; DOAN17; FAN21; XIE22; HASS18; HASC23]. Performance metrics such as reachability, manipulability, torque, and energy consumption receive the most attention if the workspace is analysed for executability [HASC23; TRAB22; DAI21; HASS18]. If multiple performance metrics are considered, the same metrics are applied to different tasks, even though different tasks may have different requirements. However, it is common for approaches to focus primarily on only one task type e.g. welding in [PELL15; PELL17; DOAN17], medical operation in [TRAB22], or coverage tasks like painting or cleaning in [DAI21; HASS18; HASS16; HASS15]. The focus on one task type restricts the general applicability to changing assembly tasks, encompassing a wide variety of tasks and operations.

Although some sources incorporate equipped end-effectors [TRAB22; MUTT21; MALH21], none have investigated the effects of changing equipped end-effectors on a) the metrics-driven representation of executability and b) the simulation and calculation of IK. Given this thesis's focus on the dynamic nature of changeable assembly and the associated variability in assembly tasks, it is reasonable to assume that end-effectors must be frequently changed to align with the specific requirements of diverse assembly tasks.

Only a few sources address multiple robots [HASC23; SUEM16; DAI21; KALA18; HASS15; HASS18; GADA17], with [HASC23; HASS15; HASS18] being the only ones considering multiple and mobile robots.

In summary, no solution is known to fulfil the requirements for the formation planning of mobile robots in changeable assembly systems. Moreover, no partial solution to building blocks can be directly applied and utilised for formation planning.

Table 3.2: Results of the qualitative literature study in the field of formation planning of mobile robots in assembly..

Author	Publication title	PF	BP	WS	AFP
LIM ET AL. (2017)	Multi-objective hybrid algorithms for layout optimization in multi-robot cellular manufacturing systems	○	●	●	●
MUTTI ET AL. (2021)	Towards optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms	○	●	●	●
Combined contributions of [PELL15; PELL17] with primary contribution PELLEGRINELLI ET AL. (2017)	Multi-robot spot-welding cells for car-body assembly: Design and motion planning	●	●	●	●
SPENSIERI ET AL. (2016)	Optimal robot placement for tasks execution	○	●	●	●
DOAN ET AL. (2017)	Optimal robot placement with consideration of redundancy problem for wrist-partitioned 6R articulated robots	○	●	●	●
SUEMITSU ET AL. (2016)	Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems	●	●	●	●
GADALETA ET AL. (2017)	Energy-optimal layout design of robotic work cells: Potential assessment on an industrial case study	○	●	●	●
FAN ET AL. (2021)	Base position optimization of mobile manipulators for machining large complex components	○	●	●	●
BANAS ET AL. (2015)	Determination of the robot location in a workcell of a flexible production line	○	●	●	○
KAWABE ET AL. (2022)	Optimal Motion Planning and Layout Design in Robotic Cellular Manufacturing Systems	○	●	●	●
XIE ET AL. (2022)	Base position planning of mobile manipulators for assembly tasks in construction environments	○	●	●	●
TRABELSI ET AL. (2022)	Optimal Multi-robot Placement Based on Capability Map for Medical Applications	○	●	●	●
ZHANG ET AL. (2020)	A novel coordinated motion planner based on capability map for autonomous mobile manipulator	○	●	●	●
Combined contributions of [MALH19b; MALH19a; MALH21] ²¹ with primary contribution [MALH21]	Automated planning for robotic layup of composite prepreg	○	●	●	●
PAUS ET AL. (2017)	A Combined Approach for Robot Placement and Coverage Path Planning for Mobile Manipulation	○	●	●	●
Combined contributions of [HASS15; HASS16; HASS18; DAI21] with primary contribution HASSAN ET AL. (2018)	Collaboration of Multiple Autonomous Industrial Robots through Optimal Base Placements	●	●	●	●
KALAWOUN ET AL. (2018) combined with KALAWOUN (2019)	Optimal robot base placements for coverage tasks	○	●	●	●
HASCHKE (2023)	Planning of Indoor Construction Tasks for Mobile Manipulators	●	●	●	●

3.3 Research deficits

Research gaps were identified and deduced upon completing the systematic literature review. Subsequently, the main deficits for formation planning and the central deficits for each building block are extracted in the following, and summarised in figure 3.3.

Formation planning. As derived in the systematic literature review, no approach combines problem formulation for task allocation, base placement and workspace analysis of mobile robots in changeable assembly systems adequately. None of the approaches comprehensively address the interdependence between task allocation, workspace evaluation, and base placement. Addressing this research deficit enables the potential for performance enhancement of changeable assembly through the implementation of an integrated solution for the formation planning of mobile robots.

Modelling and problem formulation for task allocation. Existing approaches primarily concentrate on placing robots for coverage tasks rather than distributed discrete tasks. While numerous techniques are proposed for coverage tasks involving homogeneous robots and task types, these methodologies need to be expanded to accommodate diverse tasks and robot types to facilitate their applicability in assembling products with high variability. Typically, the focus is on handling objects with simple geometries, and the scope of the problem often remains confined to a single robot's workspace. Many studies emphasise optimising local or global performance metrics for robot base placement, overlooking collision avoidance and neglecting task-specific performance metrics. Currently, there is lacking consideration of the heterogeneity of robots, end-effectors and tasks to be applicable to changeable assembly systems. Consequently, the approaches fail to guarantee the executability of assigned assembly tasks. Therefore, a problem formulation and consistent semantic modelling of the mobile robots and their capabilities has yet to be developed.

Workspace representation and evaluation of executability. Adjusting an assembly station to variable tasks necessitates constant changes of equipped end-effectors on robots. The robot can only execute tasks, if equipped with end-effectors. End-effectors significantly impact a robots' range and performance metrics, in dependence on the end-effectors dimensions and the added redundancy of degrees of freedom. Currently, end-effectors are not considered and their effect on the robot workspace is ignored. Various criteria and methods for workspace analysis of industrial robots are available in the literature, but performance metrics are not chosen in dependence on or tailored to specific task types. Currently, no method exists to determine which metric should be applied to evaluate a robot's capability for a specific task [RUSS22, p.13]. Additionally, no workspace representation exists that enables the evaluation of the workspace concerning multiple metrics tailored to the task.

Base placement for multiple mobile robots. A research gap exists in understanding optimised base placement strategies for coordinating multiple robots in scenarios involving distributed task poses of different task types (coverage tasks as well as discrete tasks). Moreover, incorporating task allocation and workspace evaluation of robots into the assignment of base placements to each robot represents a research gap. In summation, present investigations overlook the inclusion of multiple heterogeneous robots, the impact of equipped end-effectors

on executability, and the consideration of various task types, particularly distributed tasks in difference to coverage tasks. Consequently, the previously identified research gaps contribute to the absence of formation planning for mobile robots in assembly scenarios.

3.4 Research scope and research question

The pragmatic goal of this thesis is to **automatically evaluate and enhance the executability of mobile robot formations in assembly**. Thereby, the overarching research scope is the realisation of the formation of mobile robots in a changeable assembly station. Figure 3.2 envisions the input and output scenarios of formation planning. An assembly station consisting of a number of resources — robot arms, end-effectors and mobile bases — is considered input. As soon as an assembly task consisting of task poses and products is assigned to the assembly station, a formation shall be planned. The output — the planned formation — determines the allocation of task poses to combined mobile robots. Each combined mobile robot shall consist of a mobile base, a robotic arm, and an end-effector equipped with the necessary capabilities to execute the allocated tasks. Therefore, the evaluation of the workspace in which each combined robot can execute tasks is required, to determine the spatial base placement of each robot.

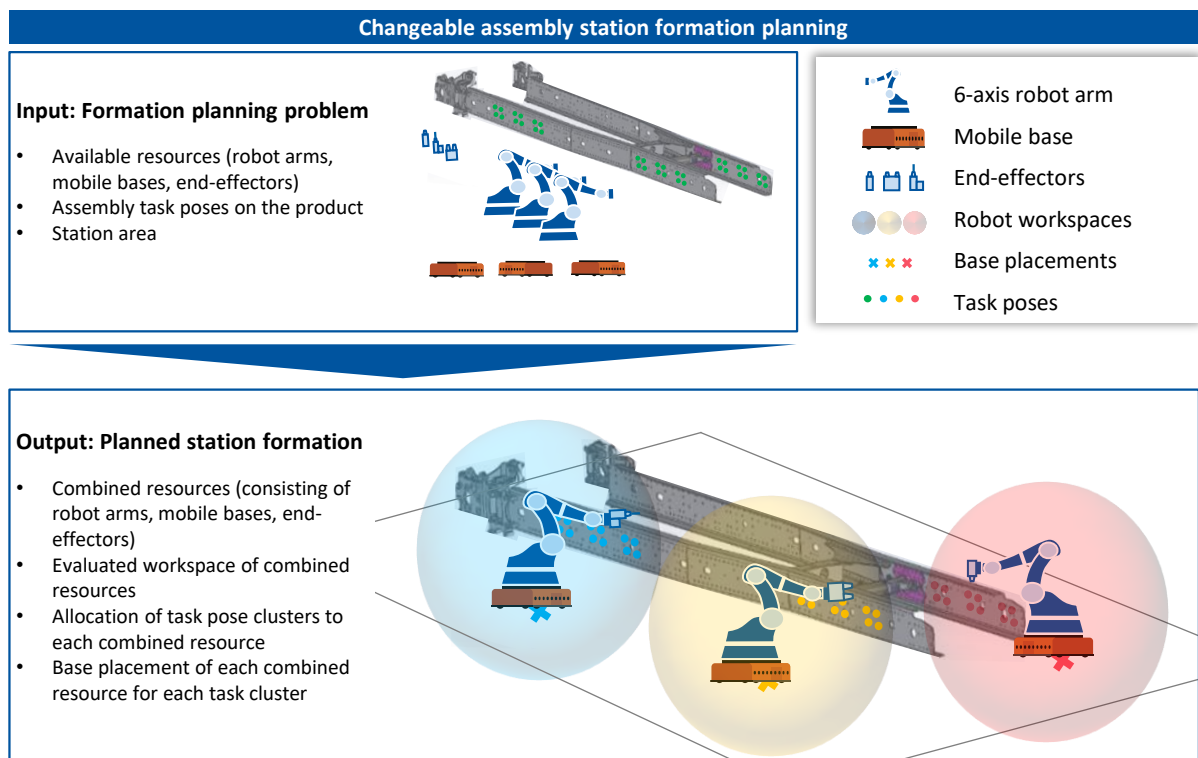


Figure 3.2: Input to the formation planning problem and a planned formation as output. The planned station formation is defined by the colour-indicated allocation of task poses to robots placed at base placements. Thereby, the robot is composed of a mobile base, a robot arm and an end-effector, providing the needed capabilities to perform the allocated tasks.

To realise the research scope of formation planning of mobile robots, the previously identified research deficits shall be investigated and answered. The primary research question guides the research conducted to reach this goal:

Primary research question: *Can capability-based formation planning of mobile assembly robots in stations of changeable assembly systems ensure the executability of the assembly tasks?*

Therefore, the following three underlying research questions are formulated, structuring the primary research into three subsequent research areas, which are mapped to the three building blocks of system modelling, executability evaluation and base placement, as visualised in figure 3.3.

1st underlying research question (URQ1): *How can multipurpose assembly robots, their capabilities and assembly tasks be modelled for a task allocation?*

2nd underlying research question (URQ2): *Can the executability evaluation of assembly tasks be extended whilst considering equipped end-effectors on mobile robots?*

3rd underlying research question (URQ3): *How can assembly tasks be clustered to plan formations with reduced base placements of mobile assembly robots?*

The key research gaps align with the three underlying research questions delineating the central research areas of this thesis. Figure 3.3 ascribes the interconnections between these gaps and the research areas, specifying the underlying research question associated with each particular gap. The three research areas are investigated iteratively, following the applied research methodology of DSR and incorporate research results from the subsequent research areas into the preceding areas. The subsequent requirements for developing artifacts are thoroughly investigated at the beginning of the affiliated chapters.

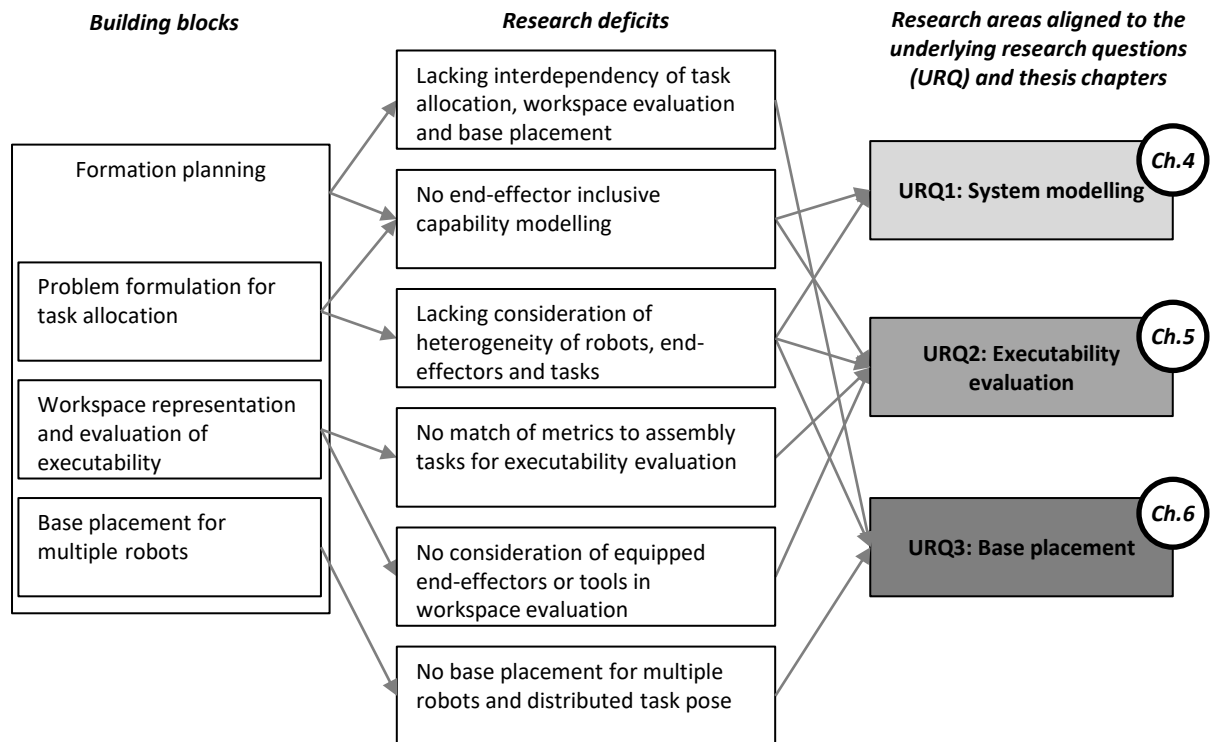


Figure 3.3: Mapping of the buildings blocks of formation planning to the identified research deficits to the underlying research questions, research areas and thesis chapters.

4 Semantic description model of mobile robots and their capabilities

"Every problem-solving effort must begin with creating a representation for the problem a problem space in which the search for the solution can take place." [SIMO96, p.108] The foundation for planning formations of mobile robots in changeable assembly is to create a model representing the complex, real world in a simplified concept. This technical capture and representation of knowledge is not only at the beginning of the system development process, but its solution is also crucial for the quality of the resulting system. [BEIE19, p.1] Therefore, the first step towards realisation is a unique description model containing identified system knowledge, answering the URQ1: ***How can multipurpose assembly robots, their capabilities and assembly tasks be modelled for a task allocation?*** The modelling is a base for the second and third design artifacts presented in chapter 5 and chapter 6, focused on creating new knowledge. Due to their foundation on precise and comprehensive syntactic and semantic rules, ontologies are apt for knowledge representation and reasoning [OLIV19, p.4]. Following the postulate of IIVARI to base DSR on an ontology [IIVA07, p.3], this chapter presents the ontology CAPILANO built for the formation planning of mobile robots.

The following chapter introduces the ontology CAPILANO as a knowledge artifact. In the beginning, the applied methodology for creating an ontology is introduced in section 4.1. Thereafter, in section 4.2, the requirements to establish a semantic description model of mobile robots and their capabilities, serving as a fundamental element in assembly station planning, are derived. Established ontologies in the manufacturing domain are compared to those requirements, highlighting the need for developing a novel ontology. Section 4.3 details the conceptualisation of the ontology based on reusable features of established meta-models by defining the semantic and syntactic modelling of the assembly resources and their capabilities. The operational implementation of CAPILANO is provided and the implementation choices are discussed. Section 4.4 details the ontology-based task allocation framework. The resulting ontology is verified and validated in section 4.5 by applying the conceptual ontology to the concrete example of the use case truck assembly. The chapter concludes with a summary and a critical reflection on the first design artifact in section 4.6.

Chronologically, the ontology was iteratively adapted in terms of DSR during the investigation and development of the subsequent artifacts. Three major cycles of model evaluation and refinement can be differentiated. In the first cycle, a conceptual ontology to allocate tasks to assembly resources by matching capabilities was developed. In the second cycle, the ontology was adapted to the requirements deriving from chapter 5. During the third cycle, information about the assembly tasks (distributed vs. continuous task) was added to the task description to adhere to the requirements occurring during the development of chapter 6. Here, only the final ontology is presented, and no differentiation between the three design cycles is made.

4.1 Applied methodology of building CAPILANO

The inconsistent utilisation of terminology, diverse structures, implementations, modelling languages, and the inherent complexity have resulted in a lack of global consensus regarding the appropriate description model or ontology to be employed under specific circumstances [BADE19, p.151]. Structuring the process of identifying reusable ontologies or description models and defining a fitting ontology requires a methodology.

4.1.1 Ontology development methodology

Various methodologies for formally designing ontologies exist in the literature, as detailed in section 2.3.3. Here, the following methodologies were applied: The first two steps of the digital twin pipeline by GÖPPERT ET AL. (2021) — the definition phase involving ontology development and the modelling phase encompassing the modelling of a use case — were incorporated. Given the pipeline's primary focus on digital twins and more general consideration of ontologies, there's a need to combine it with methodologies dedicated to developing ontologies to create the desired domain ontology. Consequently, the established methods "**Ontology Development 101**" of NOY ET AL. (2001) and "**Ontology Engineering Methodology**" by SURE ET AL. (2009) were followed during the ontology development. As each of the three methods emphasises different aspects and applications but does not cover all essential steps and phases in sufficient detail, a comprehensive modelling approach can be created by integrating them. Figure 4.1 provides an overview of the resulting methodology as applied for developing CAPILANO. The index of each step indicates its origin in the combined methodologies.

The first steps constitute the definition of the domain and scope of the ontology, followed by a study of need and feasibility. During the ontology-based definition, the domain ontology is conceptualised. Therefore, reusable features of models and ontologies are identified. All three combined methodologies have the two steps of assessing the literature and potentially reusing ontologies in common. NOY AND MCGUINNESS (2001) and SURE ET AL. (2009) propose the definition of requirements as preceding step. Here, the requirements and related competency questions are presented before analysing the fulfilment of the criteria of existing ontologies. During refinement, the identified reusable features are applied to constitute the classes, hierarchy and taxonomy, resulting in the conceptual domain ontology. By modelling concrete instances of a use case according to the standard provided by the domain ontology, an application ontology results [LOVR08, p.3]. Both the ontology-based definition and the standardised modelling inherently necessitate iteration through the evaluation of the resulting ontologies regarding the defined requirements. Following the modelling, the ontology can be deployed and utilised in its application environment.

4.1.2 Application of the methodology

The ontology's domain and scope were shaped by the research emphasis of the IoP excellence cluster and are elaborated in section 1.2. In the context of conducting research, the necessity of a description model of the domain emerged.

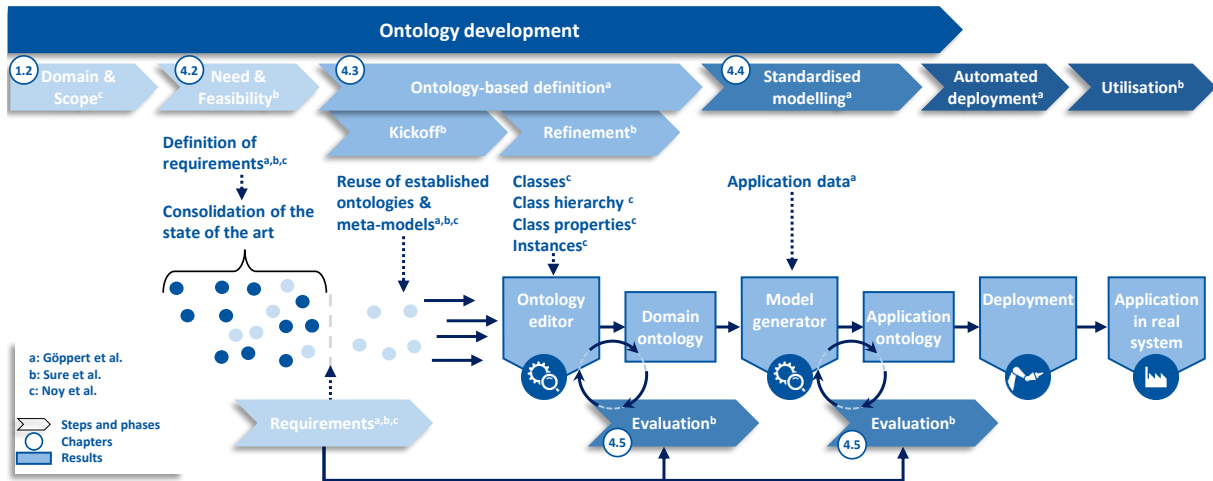


Figure 4.1: Mapping of the phases of ontology development, combined from [GÖPP21; NOY01; SURE09], to thesis chapters.

The subsequent need and feasibility study is based on discussions with the "Ontology-expert group" associated with the IoP research project to confirm the viability of utilising ontologies as a description model. The ontology's scope was further detailed, and general requirements were outlined in discussions again with the "Ontology-expert group". Specific requirements underwent refinement in workshops with the "Future Assembly" research group of the IoP. These resulting requirements and competency questions are detailed in section 4.2.1. A consolidation of state of the art was conducted to identify applicable ontologies and meta-models, as presented in section 2.3 and focused on formation planning in section 4.2.2.

Section 4.3 describes the findings of the ontology-based definition phase resulting in the domain ontology and task allocation framework CAPILANO. Reused features of existing ontologies and meta-models are detailed. Sections 4.3.1 and 4.3.2 present the consecutive refinement of the class models and hierarchy, following steps three to seven of NOY AND MCGUINNESS (2001).

A use-case-specific description model — an application ontology — was created during the standardised modelling phase. Moving to the evaluation phase, the proposed ontological design was evaluated in an assembly scenario within a changeable assembly system (see section 4.3.3). Verification and validation processes, outlined by SARGENT (2010) and GÓMEZ-PÉREZ (2004), involved application ontology and performance testing, results of which are summarised in section 4.5.

Looking ahead, the future deployment phase outlined by GÖPPERT, GRAHN, RACHNER, GRUNERT, HORT, AND SCHMITT (2021) emphasises application and evolution, akin to SURE ET AL. (2009), which remains a future and ongoing endeavour applied to industrial use cases.

4.2 Need and feasibility phase

4.2.1 Requirements and competency questions

Beginning the definition phase of the combined methodology constitutes defining the application domain and listing requirements for the description model. The application domain was already defined in chapters 1 to 3. Following NOY AND MCGUINNESS (2001), the requirements are detailed as formulated competency questions. Thereby, the formulated competency questions address the previously formulated research gaps in section 3.4 for the description model.

Prerequisites and constraints — denoted as **General Prerequisites (GP)** — were defined for the **formal description (GP1)** and digital representation of knowledge about resources and their capabilities, as outlined in [FROS22, p.3-5]. All modelled resources have to be **identifiable (GP2)**. Additionally, adhering to changeable assembly scenarios and according to the goals postulated for the control of production processes in the IoP the description model has to be **scalable (GP3)** and **adaptable (GP4)** to new scenarios [PENN19b, p.7; GINS22, p.6455]. Hereby, scalability refers to adjusting an assembly station to a change in production requirements (e.g. increase or decrease of production, change in products etc.) by excluding or including additional resources without causing disruptions of the assembly (following [HOFF17, p.50; SPIC02, p.5]). Moreover, the **extendability (GP5)** to novel resources, represented by classes and concepts, into the assembly station has to be foreseen.

More explicitly aimed at the goal of formation planning, the following requirements — denoted as **Specific Prerequisites (SP)** — are formulated. The modules requiring input from a description model are the ontology-based task allocation, the evaluation of executability and the base placement of the robots. To achieve transferability and generalisability, the **capability modelling** should be technology-, product-, process- and resource-independent, analogously to the requirements derived by [MALA18, p.70; WESE20]. When applied, the capabilities should be extended with **parameters (SP1)** (e.g. description of limits or dependencies of the capability → "Transporting" with a max. weight of 20 kg). As robotic systems are modular and built up of different components (as depicted in figure 2.2), the capability modelling should reflect their modular nature. The resources only exhibit certain capabilities when specific resources are combined: a screwdriver that is not equipped with a robot arm cannot screw screws, as well as a robot arm without a screwdriver cannot screw. Thus, the capabilities must be **inherited (SP2)** by the combined resources, and if all **conditions (SP3)** (e.g. interface compatibility) are met, the combined resource exhibits the capabilities.

Additionally, it should facilitate queries for assembly resources and their capabilities, **matching resources to assembly tasks (SP4)** based on availability and requested capabilities as derived in [KLUG22c]. Furthermore, to enable more advanced and complex task allocation, there must be a provision for **transferring query results to third-party software (SP5)**. For effective station control and task allocation, it is essential to include **time-dependent properties (SP6)** and updates thereof, such as whether a resource is idle or not idle.

The robots' base placement, as described in chapter 6, depends on an Unified Robot Description Format (URDF) model of the robots and needs a .csv file of the assembly task poses as input.

It is assumed that the .csv file of the assembly task poses is ever-changing and will be given it is not further considered here. The calculation of metrics, as described in chapter 5, is based on an URDF model of the robot as well as a Computer-Aided Design (CAD) model and dimensions of the equipped end-effector. As URDF and CAD are established and often applied file formats, their content shall not be duplicated; instead, the **reference to their respective external files (SP7)** must be given. **Resource dimensions (SP8)** shall be modelled.

Based on iteration during the design cycle following the DSR methodology, and after thoroughly analysing requirements and considering potential use cases, the following competency questions for CAPILANO are formulated:

- CQ01 Which capabilities does a (combined) resource have? (e.g. transporting, screwing)
- CQ02 Which capability parameters does a (combined) resource have? (e.g. transporting a maximum of 25 kg)
- CQ03 Which resource is compatible with which other resource?
- CQ04 Which combination of resources does provide which capabilities?
- CQ05 Which combination of resources provides capabilities with a certain parameter threshold? (e.g. transporting an object with a weight of 125 kg)
- CQ06 Which transformations of the current status of the resources are needed to transfer into the desired one? (e.g. which robot has to unequip which end-effector and which resource has to equip which end-effector?)
- CQ07 What are the current states of the resources? (e.g. idle, equipped, out of order)
- CQ08 How many degrees of redundancy does the end-effector have?
- CQ09 What are the dimensions of the end-effector?
- CQ10 Where is the model (URDF, CAD or alike) stored?

During the evaluation phase, the formulated requirements as well as the competency questions, are applied to assess whether the developed ontology meets the requirements.

4.2.2 Consolidation of the state of the art

Based on the derived requirements, the reusability of established ontologies shall be investigated in the following. First, an analysis of existing ontologies is presented, highlighting the need for a novel ontology to fulfil all previously derived requirements. For the literature analysis, the criteria explained in table 4.1 shall be applied. Table 4.2 then overviews ontologies regarding their fulfilment of the criteria indicating applicability for formation planning. The ontologies were selected based on the investigation for section 2.3 and the pre-analysis of applicable and fit ontologies in related works [GÖPP21; MÜNK21; PEDE16; SCHO18; FROS22; BUCK22].

The INstitute of Electrical and Electronics Engineers (IEEE) standard Core Ontology for Robotics and Automation (CORA) argues for the need for and provides an upper-level ontology for a formal reference vocabulary representing knowledge about robots [ROBO15, p.ii]. The objective is to simplify the integration of robots and data, ultimately enhancing their transferability and compatibility through a standardised representation and reasoning of knowledge about robots and automation. CORA is based on the vocabulary introduced in Suggested Upper Merged

Table 4.1: Evaluation criteria for the suitability of ontologies and description models for formation planning.

Evaluation criteria	Fulfilment level
Fit to the intended application domain and scope	○ None
	◐ Focus on assembly or robots (set as a requirement to be considered here)
	◑ Mobile resources
	◒ Combinable resources (e.g. mobile base with robot arm)
	● Consideration of resource state (e.g. idle)
Capability description	○ None
	◐ Capabilities are considered
	◑ Enriched with parameters (e.g. maximum payload)
	◒ Inheritable capabilities (by combining resources to a combined resource, the capabilities should be inherited)
	● Conditional capabilities (when combining resources, capabilities might result that either partial resource did not exhibit)
Task allocation through matching of required and proposed capabilities	○ None
	◐ List of resources fulfilling the requirements
	◑ List of different combined resources fulfilling the requirements
	◒ Ranked list of resources fulfilling the requirements
	● Optimised task allocation (e.g. equalising distribution of tasks, including path lengths, end-effector changing times, etc.)

Table 4.2: Suitability of ontologies and description models for formation planning of mobile, multipurpose assembly resources. The Harvey Balls indicate criteria fulfilment.

Source	Ontology	Title	Scope	Capabilities	Task allocation
IEEE Standard Ontology (2015)	CORA	IEEE Standard Ontologies for Robotics and Automation	◒	○	○
Combined approach of PEDERSEN ET AL. (2016) and its extension in SCHOU ET AL. (2018)	—	Skill-based instruction of collaborative robots in industrial settings	◒	◑	○
JÄRVENPÄÄ ET AL. (2019)	MaRCO	The development of an ontology for describing the capabilities of manufacturing resources	◒	●	◑
PERZYLO ET AL. (2019)	BaSys 4.0 ontology	Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective	◒	◒	◑
WESER ET AL. (2020)	C4I	An Ontology-based Metamodel for Capability Descriptions	◐	◑	○
MÜNKER ET AL. (2021)	IoC-OSO	Online capability-based resource allocation for on-site construction operations utilizing digital twin models	◑	◑	◑
GINSTE ET AL. (2022)	—	A formal skill model to enable reconfigurable assembly systems	◒	◑	◑
VIEIRA DA SILVA ET AL. (2023)	—	A capability and skill model for heterogeneous autonomous robots	◑	◑	◑

Ontology (SUMO)²³. As others have denoted, CORA is too general to apply to complex assembly scenarios directly [OLIV19, p.17]. Additionally, capabilities are not considered.

Manufacturing Resource Capability Ontology (MaRCO) formally models resource capabilities in the manufacturing domain. Its objective is to establish a vendor-independent representation of resource capabilities, managing them during the resource lifecycle and matching required and offered capabilities. [JÄRV19, p.963] Following the PPR modelling approach, they extend the concept with resource capabilities by considering elemental capabilities, building up combined capabilities and combining resources leading to their combined capabilities. The authors themselves criticise the incompleteness of the capability knowledge base as well as the applied form of matchmaking between resources and tasks [JÄRV19, p.977].

Within the BaSys 4.0 ontology, modular resources offer a combination of slave capabilities coordinated by master capabilities. BaSys 4.0 incorporates querying by aligning requested capabilities with those available, while it does not encompass the allocation of tasks to individual resources. [PERZ19] WESER ET AL. (2020) continue the works of BaSys 4.0 and argue for a capability-based description, easing the implementation of new resources, task allocation and purchases of new technologies. Therefore, they propose the ontology Capability For Industry (C4I) as an upper-level ontology modelling and defining an assembly resources capabilities. Even though capability-based matching and task allocation are given as motivation, they are not detailed.

MÜNKER ET AL. (2021) focus on capability-based allocation of tasks to resources in the construction domain. The end-to-end pipeline of modelling digital twins is applied (referring to [GÖPP21]). The modelling of resources and their capabilities only considers stand-alone resources and no combination of parts (end-effector, robot arm, etc.) to create combined resources. Additionally, the task allocation matches tasks with resources based on a matching of required and provided capabilities and does not consider additional criteria, like the closest match of the ability properties (e.g. a robot able to lift 50 kg should not be taken to lift an object of 2 kg, if a robot that can lift 5 kg is also available).

GINSTE ET AL. (2022) focus on combined offline and online planning of resources. They propose a capability model based on the interface state changes (assembled or not assembled) of the product parts during assembly and the feedback of this information during execution.

VIEIRA DA SILVA ET AL. (2023) focus on connecting the hardware- and implementation-independent modelling of capabilities with the execution of these. Additionally, they consider mobile robots in a more general term, thus including search and rescue or investigating robots, like Unmanned Aerial Vehicles (UAVs) and water robots [VIEI23, p.145]. Therefore, their modelling is aiming at a broader scope than mobile robots in assembly and is thus too general. Additionally, the inheritance of capabilities, combining resources and task allocation, is not considered, even though it can be argued that one can query for the modelled capabilities and thus provide task allocation in a basic format.

²³The SUMO ontology serves as a high-level ontological framework, establishing fundamental ontological categories that span across all domains. [NILE01]

As can be seen from the comparison of ontologies, no single ontology fulfils all criteria suitable for use as a description and knowledge base of heterogeneous, multi-purpose resources in changeable assembly systems. Established querying methods focus on providing a feasible combination of resources to provide a requested capability. Still, they cannot address more complex problems, such as including travel costs or considering the status of the resources. Additionally, many ontologies are not publicly available. Furthermore, they differ in their level of granularity, implementations and modelling languages, making it difficult to merge and reuse them. Therefore, the need arises to create an ontology designed for the purpose at hand: formation planning of mobile robots.

The intended reuse of specific features and parts of ontologies to create a novel, domain-specific ontology for formation planning is detailed in the following. The adapted parts are highlighted during the development of the resource and capability model.

4.3 Ontology-based definition

The following section details the ontology-based definition phase and details the development of a semantic description model of mobile robots and their capabilities. The goal is to represent the totality of all data and features that characterise an associated asset or represent important information for other assets. [NORM16, p.18] The conceptual CAPILANO based on [KLUG22b] is presented, describing heterogeneous assembly resources and their (combined) capabilities uniformly and formally. It can thus be used as a basis for a capability-based task allocation to assembly resources. The development of the ontology-based task assignment framework, including the ontology taxonomy, query, and task assignment, is discussed.

An essential requirement for the automation of assembly management involves evaluating whether a given resource has the capability to carry out specific activities. The representation of capabilities in a formal model allows one to reason which configuration ($\hat{=}$ combination of robot components and periphery, like robot arm, robot base, end-effector) allows executing the said task.

The data employed for capability-based resource allocation in assembly operations encompasses three central elements: (1) the process chart, (2) part and product models, and (3) resource profiles. These data components represent distinct use case entities that require standardised modelling. The necessary parts for a complete description of task allocation are divided into the process chart, which contains a description of the processes or products, consisting of the quantity of parts and components and the assembly sequence and necessary capabilities for assembly, as well as the domain-specific ontology of resources consisting of the capabilities of the resources and their spatial dimensions.

Figure 4.2 visualises the incorporated ontologies and meta-models and for which part of CAPILANO some of their features were considered. The adapted features and the resulting modelling in CAPILANO are detailed in the following.

As proposed by GÖPPERT ET AL. (2021) and showcased in [MÜNK21], the meta-models delineating the fundamental and consistent structures of the description model and syntax for

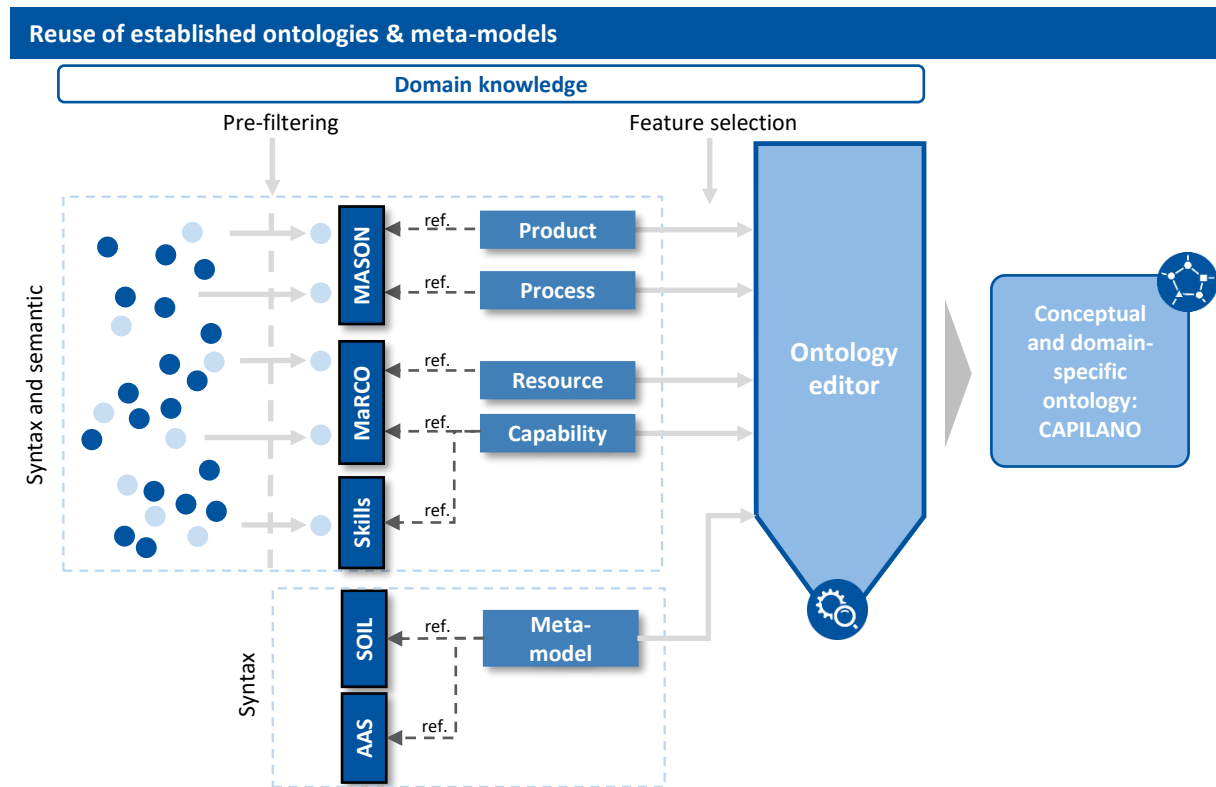


Figure 4.2: Reuse and influence of established ontologies, meta-models, and concepts onto the development of the CAPAbLiTy-based resource AllocationN Ontology (CAPILANO). The methodology is adapted from [GÖPP21].

execution Asset Administration Shell (AAS) and SensOr Interfacing Language (SOIL) are employed. AAS depicts the digital representation of real and virtual assets, communicating through a standardised communication interface. This encompasses the asset's features, characteristics, properties, statuses, parameters, measurement data, and capabilities. [HUAN21; TANT17] SOIL is originally designed for sensor systems, but the underlying meta-model can be applied to other domains, such as assembly systems. The modelled assets in SOIL are (1) components, (2) functions, (3) parameters, and (4) measurements. [BODE21] Here, the more general adaption of SOIL (in [GÖPP21]) — exhibiting the asset of variables instead of measurements — shall be adapted. Figure 4.3 provides the resulting generic description of a CAPILANO component class as well as two class examples, which will be detailed in the next two sections 4.3.1 and 4.3.2. The components serve as the structural representation of an asset, encompassing any number of child elements. Therefore, resources as components can be modelled consisting of one or more other resources, thereby contributing to the answerability of CQ01–CQ06. Functions are composed of a set of input arguments and a set of return parameters. The variables represent dynamically changing information (e.g., State, Location or Configuration), while parameters encapsulate static data (e.g., maximum payload, maximum velocity). Each component class is affiliated with an AAS containing the meta-structure of AAS, encompassing essential elements such as ID, Type, Name, Variables, Parameters, Components, and more.

Following VAN BRUSSEL ET AL.; MARTIN ET AL.; CUTTING-DECELLE ET AL.; CAO ET AL., PPR modelling is taken as a base. Enhancing flexibility and adaptability, the direct

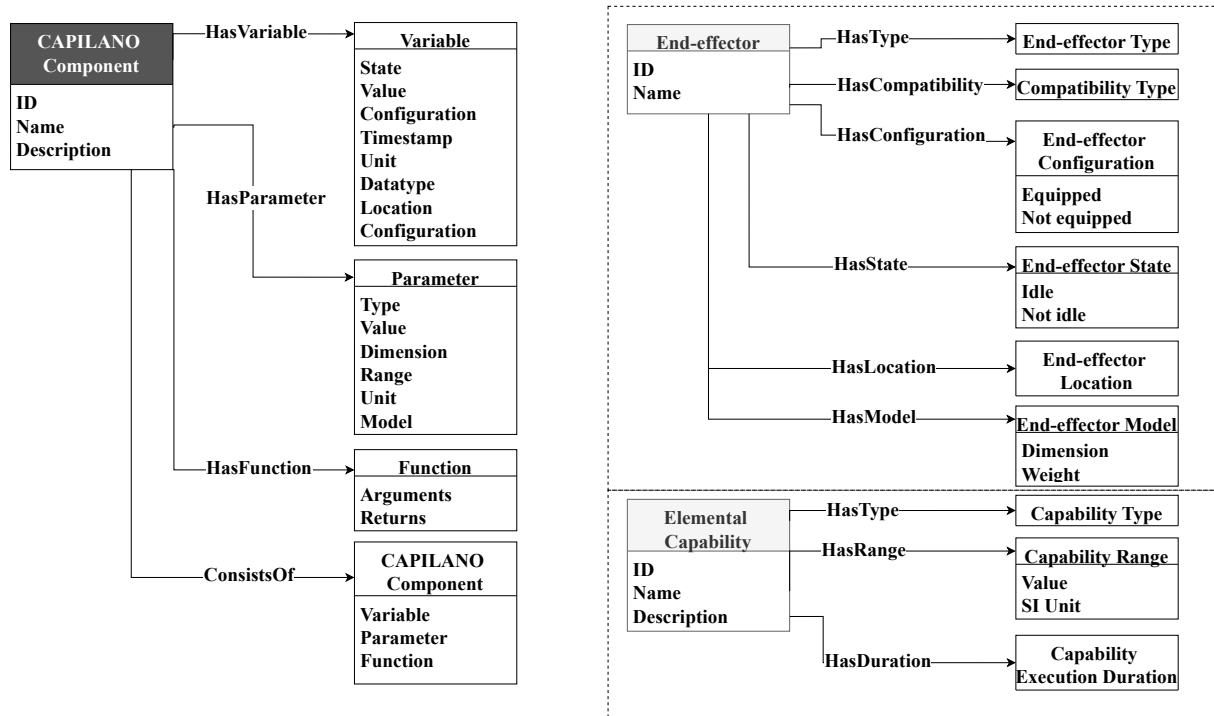


Figure 4.3: Generic CAPABILITY-based resource AllocationN Ontology (CAPILANO) class model with two specific class model examples: the Elemental Capability and the End-effector classes.

connection of PPR is decoupled by modelling capabilities as a connecting element [FROS22, p.2; WESE20, p.1679]. The concept of capabilities is based on and extends the works of MaRCO [JÄRV19] and the "skill" concept [BØGH12; PEDE16; SCHO18]. The selected ontologies and meta-models are summarised and referenced in section 2.3.

The resulting class model of CAPILANO in Unified Modeling Language (UML) notion exhibiting the central classes of resources, capabilities, process and product is detailed in C in figure C.1.

4.3.1 Modelling of assembly resources

Following Referenzarchitekturmodell der Industrie 4.0 (RAMI4.0) [NORM16, p.5–10] and PROSA [VAN 98], resources are structured analogously to assets or holons and thus have a defined boundary, can be built up from other distinguishable resources or combined into such, and can be assigned a value and a purpose. In terms of RAMI4.0, resources are assumed here to be Industrie4.0 components, i.e. an entity (clearly identifiable, represented and known asset) with an administrative shell and communication capability [NORM16, p.13]. In contrast to RAMI4.0 and PROSA, the considered application is focused on formation planning in assembly. Therefore, only the assembly resources robots, end-effectors, sensors and humans are considered. Thereby, the focus is on robots and end-effectors as they are executing the automated assembly operations.

Resources can be composed of identifiable other resources that can be combined to create new resources and exhibit properties and values. The utilisation of specific parameters drew inspiration from MASON, PROSA, and Perception and Manipulation Knowledge (PMK). The current static and dynamic parameters were fine-tuned based on MASON, while the interplay

of these parameters with capabilities was adapted from PROSA and PMK. [VAN 98; DIAB19; LEMA06]

The assumption is made that factors such as computing power availability, a communication architecture²⁴, and data reception through wireless communication are inherently present, and hence, do not require additional consideration in the formation planning process. Nonetheless, they are represented in figure C.1 as cyber resources.

As Physical Resources Sensors, Humans, Robots, Multipurpose Resources and End-effectors are modelled as classes. A definition of the terminology can be found in section 2.1.2. The class Robots is further differentiated into Robot Arms, Mobile Bases and Mobile Manipulators. Robot Arms as a class include different types of robots, from industrial, reprogrammable, PUMA, and Selective Compliance Assembly Robot Arm (SCARA) robots. Mobile Manipulators consist of a Robot Arms and a Mobile Base. Multipurpose Resources consist, as visualised in figure 2.2, of a Mobile Base, a Robot Arm and an End-effector. One example of the resource class End-effector is shown in figure 4.3. Physical Resources provide inheritable capabilities to the Physical Resources they can be combined with. The form modelling of resources directly contributes to the answerability of CQ03, CQ07, CQ08, CQ09, CQ10, and indirectly contributes to CQ01, CQ02, CQ04, CQ05, and CQ06, through the combinability of resources, consideration of states, and linkage to capabilities.

4.3.2 Modelling of robot capabilities

One barrier to employing mobile robot systems remains to assign tasks to robots whilst considering the components the robot needs to exhibit the needed capabilities for those tasks [RIZK19, p.21]. In the course of the conducted research, the modelling of robot capabilities is twofold: In the following section, the semantic description, which is included in the CAPILANO, is covered. In the next chapter 5, the concept of capabilities will be re-iterated and extended by considering quantitative metrics evaluating the spatial distribution of capabilities of robots in their workspaces.

Modelling capabilities aims to allocate assembly tasks to assembly resources that have the capabilities required to perform them. A necessary prerequisite is the decomposition of product assembly plans into precisely these elementary operations, which can be equated with capabilities. If given, the assembly tasks can be allocated to the resources. In the following, the modelling of the assembly task execution capabilities of assembly resources is designed to provide generic descriptions of provided and required capabilities linking PPR.

The notion of capabilities facilitates the hardware-agnostic description of processes and their related tasks. The required robot capabilities for product assembly depend on and are determined by the product's design. The capabilities semantically abstract the functionality of a resource to cause an effect on a part during a process step in a specific assembly context. The effect can be described by the change in property values of the object, i.e. the part or product.

²⁴A potential communication architecture is presented by BUCKHORST (2023).

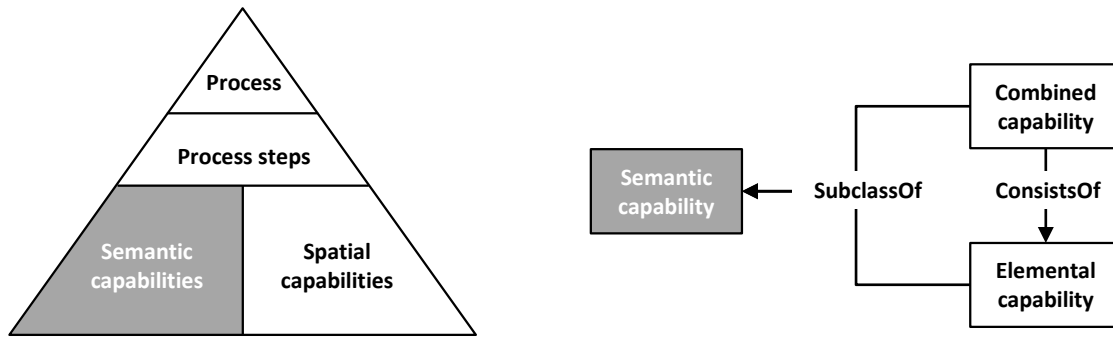
Entities such as the VDI, the IEEE or the Deutsches Institut für Normung e.V. (DIN) furnish classification systems (taxonomies) specific to various domains, offering a standardised vocabulary and structure with a dedicated focus on practical application in specified domains. Due to the solution-neutral and hardware-agnostic description, these standards can be used as a valid classification of capabilities [KLUG11, p.78]. Therefore, the established norms and manufacturing process standards shall be investigated for an initial classification and capability modelling.

Assembly is the process of combining individual parts to form a sub-unit, a unit, a structural group, a machine, or a final product [CIRP20, p.1; BAUE20, p.4]. Therefore, most assembly tasks can be classified as either handling or joining processes, combined with aiding operations like inspection or marking [LOTT12, p.2]. Creating a list of assembly tasks, the assembly operations in LOTTER ET AL. (2012) are extended with the tasks listed in VDI 2860²⁵, DIN 8580 and DIN 8593 [DEUT22; DEUT03; VERE90]. VDI 2860 [VERE90], DIN 8580 [DEUT22], and DIN 8593 Deutsches Institut für Normung e.V., Beuth Verlag GmbH provide guidelines to define and categorise different types of tasks. In [DEUT22], manufacturing processes are categorised into primary shaping, forming, dividing, joining, coating, and changing material properties. [DEUT22, p.11-15] Here, the focus lies on joining.

Following PEDERSEN ET AL. and WESER ET AL. capabilities are structured into levels, as depicted in figure 4.4 a). In contrast to the literature, four levels are distinguished here: The highest level is built by processes, which can be divided into the second level of process steps, which require semantic and spatial capabilities to be executed. The set of all assembly tasks for product completion corresponds to an assembly process. A process can be divided into individual process steps. Each process step consists of assembly task poses, which require a resource with certain semantic capabilities to be executed. In turn, the resources offer semantic capabilities. Thereby, process steps can be connected to resources by matching the semantic capabilities. The notion of semantic and spatial capabilities differentiates between the semantic expression (e.g. "welding") and the spatial distribution of performance metrics corresponding to this capability in the robot workspace. The spatial distribution and connection of performance metrics to semantic capabilities will be discussed in the following chapter 5.

The semantic capabilities are distinguished into elemental and combined semantic capabilities. Elemental capabilities are standalone, while combined capabilities require more than one elemental capability. A combined capability is a combination of multiple elemental capabilities and inherits the parameters of the elemental capabilities as shown in figure 4.4 b). To provide a specific combined capability, a robot must be equipped with certain tools and periphery, e.g., to transport, the robot arm must be placed on a mobile base. Transporting is defined to be an elemental capability exhibited by a resource, whilst Screwing is a combined capability that requires the combination of a robot arm with an end-effector for its functioning. The compatibility prerequisites to connect resources are integrated into the ontology by assigning Compatibility Types to resources, which can be requested, as shown in figure 4.7. Therefore, CQ03 can be answered. The instances of the resource class either inherently provide capabilities

²⁵The guideline VDI 2860 (1982) was withdrawn in 2018. Since no replacement exists, VDI 2860 is still referred to.



a) Hierarchic levels of representing processes as process steps as capabilities. By combining capabilities, process steps are defined. By combining process steps, processes are defined. Capabilities are defined through their semantic and spatial properties, whereby the spatial properties are detailed in the next chapter.

b) Elemental and combined capabilities as subclasses of semantic capabilities – class diagram.

Figure 4.4: Modelling of semantic robot capabilities.

or only provide capabilities if combined with other components: E.g. a component Robot Arm needs the tool component Welding Gun to provide the capability of Welding. This level of abstraction enables the seamless exchange of resources in an assembly station, provided they offer the same type of capabilities. The classes containing the capabilities are assigned to resource classes through class restrictions, similarly to MaRCO and C4I. Thereby, considering the compatibility of the CAPILANO resource class CQ01–CQ04 can be answered.

For the capabilities to be applied in an application scenario, they must be enriched with additional parameters [SCHO18, p.74], e.g. the range of applicability of the capability [MÜLL15, p.556]. For instance, to model the Screwing capability, its torque range — e.g. values spanning 50 Nm to 200 Nm — must be recorded. Therefore, for each capability, the necessity of a range and the unit thereof is checked and included in the modelling. Consequently, during the modelling process, each capability undergoes scrutiny for the imperative inclusion of a range and its corresponding SI (Système international d'unités) unit. Thus, CQ02 and CQ05 become answerable.

The inheritance relation of a taxonomy of the ontology can be used to search for specific types of capabilities or components with specific properties, e.g. all grippers or all resources with the capability of Transporting. This is applied in section 4.4 to perform an ontology-based assignment of assembly tasks to resources.

4.3.3 Implementing CAPILANO

The proposed model defines the syntax and semantics of the ontology and is independent of technology. The conceptual ontology is constructed in the model generator Protégé by utilising the Web Ontology Language (OWL). The OWL is a recognised and widely accepted standard endorsed by the World Wide Web Consortium (W3C) for modelling ontologies. The use of an OWL representation allows CAPILANO to build on existing inference mechanisms for OWL reasoning. OWL is a formal knowledge representation language, making it machine-

interpretable. This foundation guarantees decidability and allows for comprehensible deductions when employing expressive reasoning systems. [WESE20, p.1680; GLIM14] OWL facilitates the extraction of implicit knowledge from explicitly defined knowledge through reasoning systems [PERZ19]. With these properties, capabilities can be automatically composed of other capabilities or arranged in a hierarchy. This formal language supports reasoning based on both semantic and syntactic rules, enabling the inheritance of capabilities from one instance to another and the composition of capabilities from various sources. The resulting framework consists of a model of heterogeneous resources and their capabilities in an ontology built in Protégé in OWL, a SPARQL Protocol and RDF Query Language (SPARQL)-based query, and consecutive and availability-based task allocation in Python.

4.4 Task allocation

Figure 4.5 emphasises the focus of this thesis — the formerly called "precondition check" of the capability-based task execution (see figure 2.9 b)) — and its steps. The steps of ontology-based combinatoric of resources and semantic precondition check are discussed in this chapter. The subsequent steps of spatial precondition check and base placement will be detailed in chapters 5 and 6.

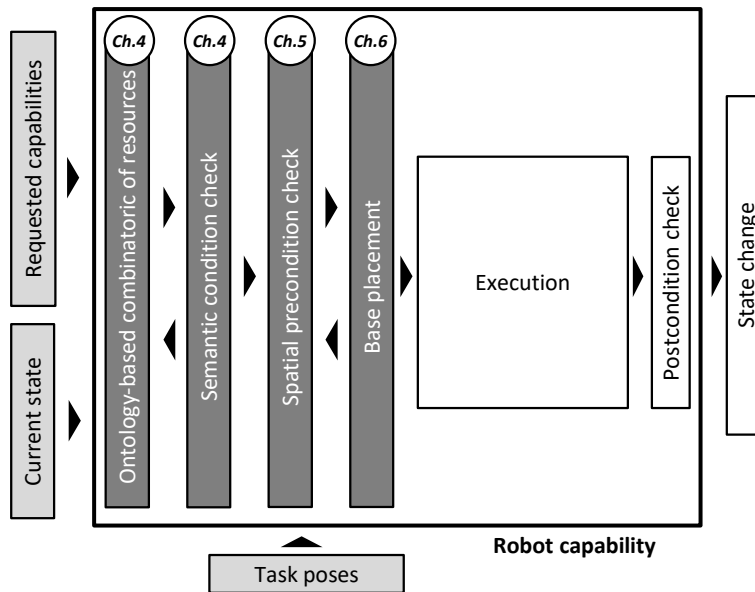


Figure 4.5: Capability-based task execution: a conceptual model of robot capabilities with precondition checks and execution planning. Based on the boundary conditions (in light grey), the precondition checks or planning steps (in dark grey) verify executability before execution. The precondition checks are mapped to thesis chapters.

As one of the primary goals of utilising an ontology to model the assembly resources and their capabilities is to enable task allocation by matching, the following will detail the task allocation framework. The conceptual framework is presented in figure 4.6. First, the information stored in the process chart and application ontology of the assembly station has to be retrieved. The process charts denote the products to be produced, the assembly sequence, and the required semantic capabilities per process step. The ontology of the assembly station contains information about the stations' resources and functions as a knowledge base populated with

the appropriate instances. The resource status (idle, battery status, combination of singular resources to combined resources), as well as the offered capabilities and their parameters, are provided in the ontology. Subsequently, the requested capabilities of the process chart are queried within the ontology. The querying results in a list of all combinations of resources providing the requested capabilities within the allowable range of parameters — if a transporting process step requires the capability of transporting 25 kg, only combined resources offering the capability of transporting with a payload of at least 25 kg are proposed. The list of all possible combinations is transferred into the next step of task allocation. Transferring the query list results into a third-party program outside of the ontology environment enables the utilisation of advanced and established task allocation algorithms. In the process of task allocation, the 'best' resource for the task is determined based on a set of defined criteria. The term 'best' is presently operationalised through the following criteria: 1) the requisite end-effector is mounted on the robot, 2) the robot is mounted on the mobile base, 3) the highest level of battery charge and 4) listing place in the query list. The combination of resources, the task allocation to those combined resources and a status update to non-idle for the execution time are transferred back into the ontology. The corresponding modelling data is updated in the ontology. This process ensures the realisation of the framework's objective to align task requirements with the capabilities of available resources. The detailed sequence of actions involved in querying and the subsequent allocation of tasks contributes to the achievement of the framework's overarching goal.

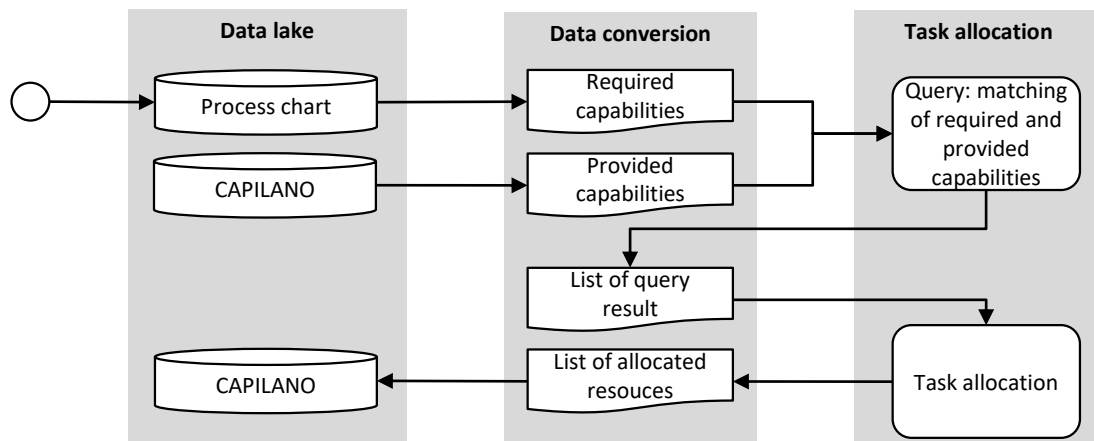


Figure 4.6: Task allocation framework including the CAPAbLiTy-based resource AllocationNtology (CAPILANO), depicted as activity diagram.

Figure 4.7 illustrates a single query loop concerning the capabilities of Screwing, Positioning, and Transporting. For the querying aspect, the specified list of required capabilities of the process chart undergoes conversion into a SPARQL query using the Python-based Cython Phaser. These SPARQL queries are transmitted to the JAVA-based HermiT Reasoner²⁶, and the resulting output is cached by evaluating the compatibility of resources to provide the requested

²⁶The HermiT Reasoner proposed by GLIMM ET AL. (2014) infers knowledge from ontologies, thereby enabling reasoning on ontologies. The reasoning is converted into consistency checks. It can be applied for consistency verification by computing the hierarchy of classes and properties. It supports SPARQL querying and can be used via a plugin in the Protégé editor. [GLIM14]

capabilities by deducing implicit capabilities. Initially, resources that possess the requested elemental capabilities are identified (ScrewDriver_1 and ScrewDriver_2 for Screwing in figure 4.7). Subsequently, the HerMiT Reasoner²⁶ examines combinable capabilities of the resources, determining, for instance, whether a specific robot, in conjunction with a particular end-effector (in this case, ScrewDriver_1 and Mobile-Manipulator via EndEffector_1), can fulfil a required capability. This mechanism facilitates the inheritance of combinatory capabilities, thereby answering CQ01, CQ04 and CQ05. In cases where multiple resources match the capabilities, they are prioritised in descending order. In the initial query loop, the resource with the closest matching parameters is selected (e.g., a requested payload of 20 kg favours a gripper with a payload of 25 kg over one with 40 kg), as depicted in figure 4.7 by a white square. Additionally, if combined resources are required, combinations that are already combined (mounted) are favoured over those that would require additional steps to be combined physically. This loop iterates until all conceivable combinations of resources are exhaustively listed in descending order. Once the Cython Phaser completes processing all queries, the cached results undergo conversion into CSV/TSV formats and are subsequently forwarded to the task allocation stage.

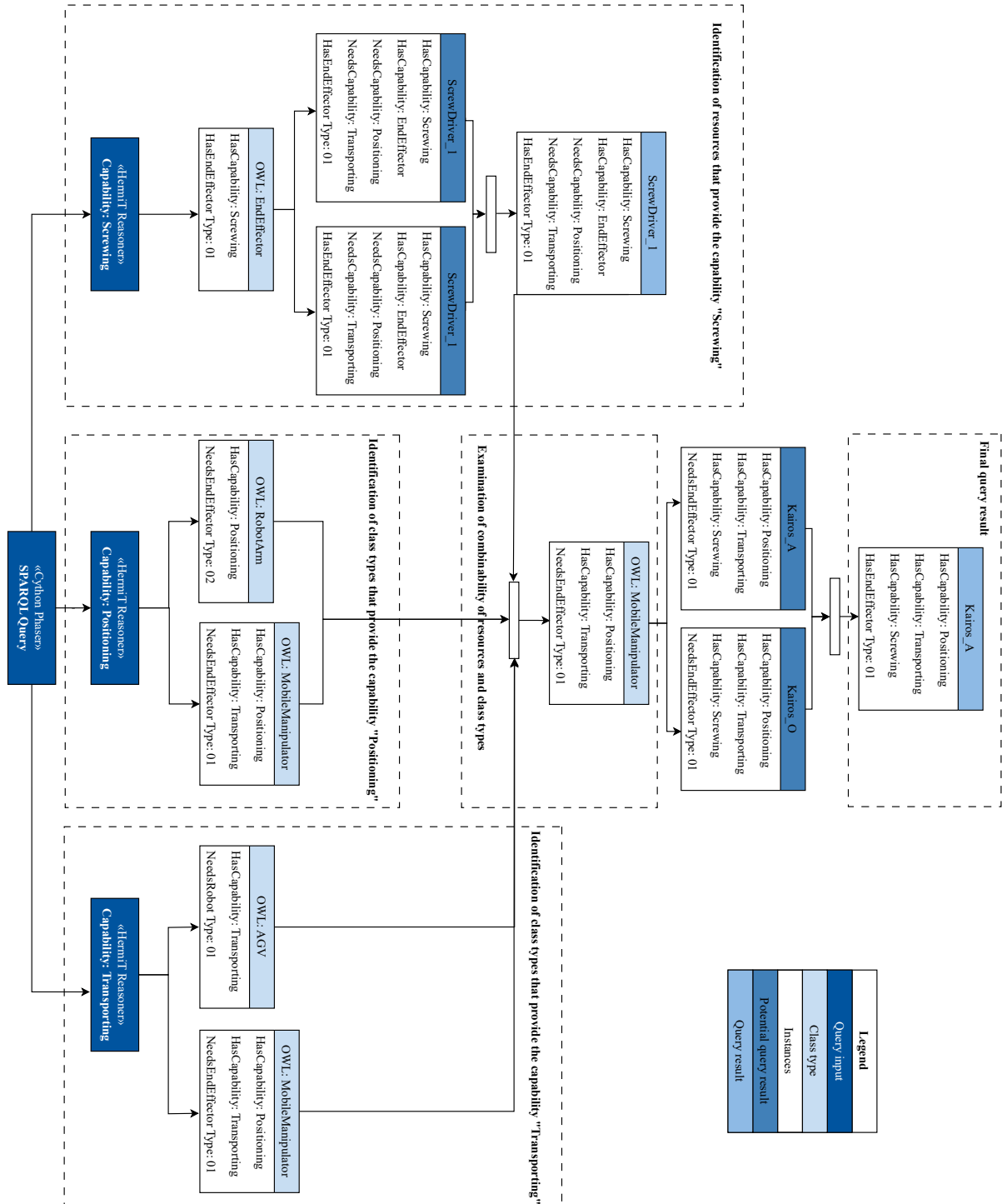


Figure 4.7: Example of the ontology querying process, depicted as activity diagram with class notation in UML.

4.5 Verification and validation of CAPILANO

The following section summarises the results of the verification and validation of the ontology. For the validation of the generated artifacts, the methodology for the validation of simulation models by SARGENT (2010) is used. According to SARGENT (2010), the validation of a model includes conceptual model validation, operation validation, computerised model validation and data validity. The underlying methodology is detailed in section 2.5.

According to SARGENT (2010), a model has to be validated for its intended use [SARG10, p.166]. Analogously GÓMEZ-PÉREZ (2004) define the evaluation of ontologies as the technical assessment of its content in relation to a set reference [GÓME04, p.255]. Therefore, CAPILANO can be validating by assessing the answerability of the initially stated requirements and competency questions (see section 4.2.1). In accordance with GÓMEZ-PÉREZ (2004), the verification and validation of CAPILANO were conducted by examining the fulfilment of evaluation criteria, specifically consistency, completeness, and expandability.

Through the iterative design cycle, which involves the alternate development of artifacts and processes and evaluation of the design results, validation occurs as an inherent part of the development of new artifacts [SARG10; OSMA97; RABE08, p.136]. Multiple verification and validation cycles were run during ontology development and iteration during the design cycles. Here, only the final evaluation is presented.

Computerised model verification

Computerised model verification checks for the correct implementation of the intended model. Analogously, “[ontology] verification refers to building the ontology correctly” [GÓME04, p.256].

An evaluation of the ontology taxonomy and a comparison between the asserted and inferred hierarchies were employed to assess the accurate implementation and programming of the conceptual ontology, encompassing computerised model verification. By automatically deducing the inferred hierarchy from the asserted hierarchy (manually implemented rules and underlying logic) of CAPILANO using the HerMiT Reasoner, the correct implementation of hierarchies, class modelling, and properties can be ensured. The HerMiT Reasoner was applied to verify the structure, machine-readability, and consistency of CAPILANO. Additionally, it is utilised to query the ontology for combined resources, providing the required capabilities. Therefore, it can be deduced that CAPILANO was built correctly, fulfilling the stated requirements.

Operational validation

An application ontology was instantiated to validate the framework within its designated scope and ascertain the acceptability of its output behaviour. Instances specific to use cases were modelled following the conceptual ontology, thereby constructing a knowledge base and a description model at an operational level, aligning with [SARG10; GÖPP21; NOY01]. The assembly process of a truck chassis served as an application scenario, wherein parts such as the Cross Member, Front Member, and Rear Member needed to be transported from storage to

the chassis and subsequently affixed to the Chassis. To execute this process, capabilities such as Screwing, Transporting, and Positioning were requested, each with distinct property parameters related to acceleration, velocity, jerk, etc. A variety of resources, including stationary robots (ABB_4600, ABB_2600), mobile robots (Kairos), and end-effectors (Gripper, Screwdriver), were available to fulfil the requested capabilities in diverse combinations.

Utilising the application ontology, an evaluation of the framework's performance is conducted by measuring the time required for task allocation processing. A set of twenty distinct queries, varying in complexity, is formulated and employed as the basis for a randomisation process to generate process charts. For each data point, five process charts, each featuring an identical number of queries but with distinct randomised content, are processed. The run-time of these five charts is averaged to ensure a consistent complexity distribution across the set of process charts. The queries and corresponding process charts are available at the following link: https://git-ce.rwth-aachen.de/wzl-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/capilano [KLUG22a].

The graphs 4.8 and 4.9 illustrate the scaling performance of the ontology. Querying was conducted for query quantities of 5, 25, 50, 100, 250, 500, 1000, and 1500, and subsequently, every 1500 queries on the same application ontology with randomised querying requests. As each query corresponds to requesting a resource for the execution of one type of assembly task, the higher input numbers are solely tested for as part of the sensitivity analysis and extreme condition testing after RABE ET AL. and SARGENT, as these numbers are not realistic for the intended scenario of task allocation for a single assembly station within the timeframe of several minutes or hours [RABE08, p.102; SARG10]. Nonetheless, it has to be noted that the resulting long computation times of around 2,45 h for an input of 1200 queries would not be acceptable in industrial applications. This is due to the transition between the SPARQL environment and Python, necessitating data conversion in both directions. Prior to implementation, enhancing the current prototype could involve optimising it through the utilisation of more efficient programming languages. The graph 4.8 indicates a linear scaling behaviour of the total runtime of task allocation based on the test data. Linear behaviour is achieved upon normalisation to an error percentage of 0.8385%. The graph 4.9 shows the variation of the average runtime per query with a maximum deviation of +0,418 % to -0,436 %. Therefore, it can be concluded that the runtime per query remains constant over scaling input queries. Consequently, it is inferred that the developed ontology displays linear scaling behaviour.

The study affirms the scalability of the developed framework, as it allows the integration of new entities for application and the inheritance of other ontologies. The depicted figures illustrate the realisation of querying and resource-task matching based on required and provided capabilities through SPARQL querying and Python-based allocation. Capabilities arising from the amalgamation of resources into a new entity can be inherited from one instance to another. Moreover, the transfer of query results to third-party software for facilitating task allocation was successfully achieved by developing interfaces that enable transfer within a Python program, demonstrating adaptability for integration with other third-party software.

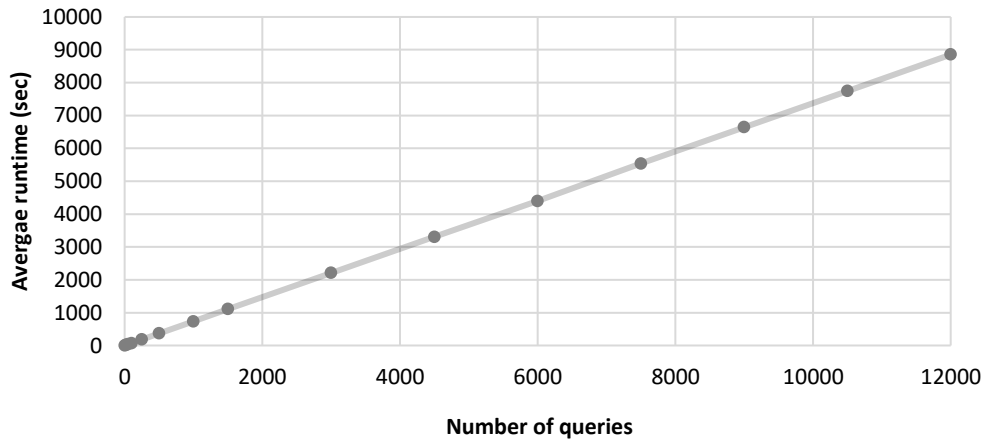


Figure 4.8: Average total runtime of the task allocation framework of CAPability-based resource AllocationN Ontology (CAPILANO) in dependence on the number of queries.

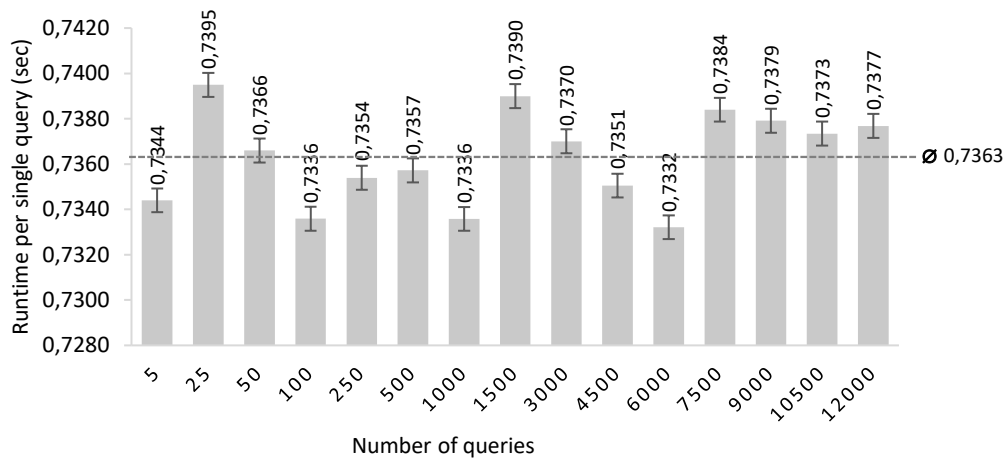


Figure 4.9: Average runtime per query in dependence on the number of total queries of the task allocation framework of CAPability-based resource AllocationN Ontology (CAPILANO).

Data validity

Section 4.5 presents the use case data utilised to instantiate the conceptual ontology. Derived directly from real-world industrial use cases within the IoP research project, the parameter data exhibits evidently validity. The application and conducted testing were discussed in the research group and reviewed during publication. Therefore, assembly experts confirmed the validity of the presented use case data, ensuring its accuracy in representing real-world systems. Additional data validity tests were not conducted.

Conceptual model validation

Model validation is the process of demonstrating that a model has acceptable accuracy within its scope of application and for the intended use [SCHL79]. Thus, a valid model can be equated with a sufficiently accurate model for the application. [OSMA97, p.135] Transferring this

general validation to ontologies results in validating that the ontology definitions model the application domain and the real world for which it was intended. [GÓME04, p.256]

This section presents a qualitative conceptual model validation of CAPILANO. The ontology's capability to answer the competency questions outlined in section 4.2.1 is assessed through instantiating in an application ontology and querying for answerability of the competency questions. The underlying application ontology is detailed in section 4.5. The competency questions represent the design requirements of the ontology. Therefore, if they can be answered, the ontology is sufficiently accurate for its intended application. Table 4.3 provides an overview of the initially stated requirements, whether it was met and with which solution the requirement was met.

Table 4.3: Matching the modelling requirements and the applied solution of CAPability-based resource Allocation Ontology (CAPILANO).

Requirement	Applied solution	Met?
GP1 Formal description	Inherently given by OWL- based modelling	✓
GP2 Identifiability of resources	Inherently considered in semantic modelling	✓
GP3 Scalability	The task allocation shows scaling behaviour.	✓
GP4 Adaptability to varying scenarios	Instances of the classes can be exchanged or adapted to novel circumstances.	✓
GP5 Extendability	Novel assembly resources or instances can be manually integrated, following the provided modelling structure.	(✓)
SP1 Parameter enriched capabilities	Considered in the capability class model.	✓
SP2 Inheritable capabilities	Inherently included through the inheritance from classes.	✓
SP3 Conditional capabilities	Precondition checking during capability matching.	✓
SP4 Matchability of resources to tasks depending on capabilities	Querying and task allocation framework.	✓
SP5 External transfer of query results	Realised in the task allocation framework.	✓
SP6 Time-dependent properties	Every Physical Resource Has a State.	(✓)
SP7 Reference to external files	Every class Physical Resource has a parameter Model with File repository (see figure 4.3).	✓
SP8 Resource dimensions	Considered in the parameter Model.	(✓)

As all requirements and competency questions are met and answered, the conceptual model CAPILANO is validated.

4.6 Interim conclusion: Description model

This chapter introduces the semantic description model of mobile robots and their capabilities CAPILANO. An ontology-based task allocation is presented. Thereby, the first underlying

research question URQ1: *How can multipurpose assembly robots, their capabilities and assembly tasks be modelled for a task allocation?* is answered.

The contribution is twofold: the **domain ontology CAPILANO** is developed, following a methodological approach, and a subsequent **task allocation framework** is presented. In contrast to alternative description models of resources and their capabilities, CAPILANO's main contribution is its capacity to model and deduce information about the combined capabilities of combined resources at a semantic level, enabling a task allocation. The chapter's contribution is summarised in figure 4.10.

Following a combined methodological ontology development approach based on established methods [SURE09; NOY01; GÖPP21], the development process of CAPILANO is detailed. During the need and feasibility study, summarised in section 4.2.2, state-of-the-art ontologies are compared to the derived requirements of an ontology as a knowledge base for formation planning of mobile resources in assembly. The analysis shows that currently established ontologies do not fulfil the requirements, thus necessitating the development of a novel domain ontology. During the ontology-based definition phase, reusable features of established ontologies and meta-models are identified and their adaption and inclusion in the novel CAPILANO are detailed. The domain ontology CAPILANO is developed in the refinement phase. During standardised modelling, the domain ontology CAPILANO is instantiated with use case data, rendering an application ontology that serves as the base for the verification and validation process. The derived ontology is added to the knowledge base as conceptual knowledge in accordance to [IIVA07, p.8] by publishing the results in KLUGE-WILKES ET AL. (2022).

The information included in the **formal representation** in CAPILANO encompasses, but is not limited to, **combinable assembly resources, inheritable resources capabilities and a capability-dependent task allocation of tasks to resources**. The proposed concept is a foundation for multi-layered capability checking at symbolic and sub-symbolic levels. It evaluates the compatibility of resources and their provided capabilities with the required capabilities to execute assembly tasks. The **semantic capabilities** of the considered assembly resources are articulated, encompassing operations such as Screwing, Transporting, or Welding enriched with specified parameters. **Capabilities are inherited** as the parts of the resources are combined; for instance, only a robot arm equipped with a screwdriver possesses the capability to screw. The **combinability of resources** is expressed by capabilities. The **task allocation** involves identifying which combination of resources satisfies the required capabilities with the closest properties match. So far, ontologies in the assembly have been limited to querying resources and their capabilities; here, the subsequent task allocation is presented as an integral part of a customised framework.

As indicated during the refinement phase and exemplified in the task allocation example, the developed ontology and framework successfully answer the initially stated competency questions (see table 4.3). The set requirements for the description model and knowledge base are met. Therefore, the first underlying research question URQ1 is successfully addressed.

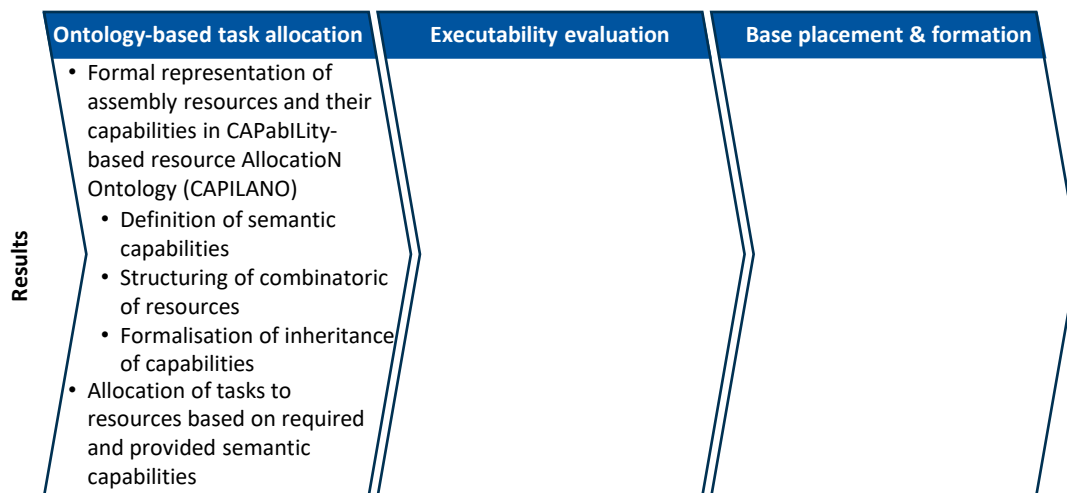


Figure 4.10: Contribution of chapter 4 to the research progress.

Critical reflection

As the real world is complex, complete representation in a model can never be guaranteed. Due to the high number of states that an assembly station consisting of mobile, multipurpose resources can assume, it is classified as complex and thus not completely describable. [ULRI81, p.8] The practical applicability of the formal capability and resource model depends on the level of detail put into implementing the model for the particular use case. The constraints include balancing the generality versus the specificity of the model and the requirement for centralised maintenance of the ontology.

Components necessary for operation, such as the modelling of environmental conditions, a communication model or a decision model, are not considered in depth here. Still, it is strongly emphasised that they are necessary.

The provided modelling of capabilities does not necessarily guarantee the robot's or resources' ability to execute and perform the actual task. The practical feasibility depends on the appropriate conditions of the situational context in which the resource is to act. For example, Transporting as a capability of an omni-wheeled mobile robot is limited to fairly flat and clean.

The modelling can be extended by incorporating the automated deployment phase of the development methodology to connect the models with the modelled physical instances, creating a digital shadow or twin.

5 Evaluation of the executability of assembly tasks by a robot

Planning feasible formations requires the placement of mobile robots in relation to the allocated task poses in such a way that all task poses are executable by the robot. To examine the executability, the second underlying research question URQ2: ***Can the executability evaluation of assembly tasks be extended whilst considering equipped end-effectors on mobile robots?*** is investigated. Executability is assumed when the task poses lie within the valid workspace. As defined in section 2.2.4, the valid workspace is defined as the area in the robot workspace wherein all task poses are reachable and executable, and the robot is unobstructed by obstacles and itself, making the placement appropriate for the intended application.

This chapter presents a method to determine the valid workspace of a robot. The semantic understanding of robot capabilities introduced in chapter 4 is extended by considering the spatial distribution of performance metrics in a robot workspace as a function of equipped end-effectors. The representation of the spatial distribution is called the performance map of a robot. Therefore, relevant metrics for specific assembly tasks are identified, and then the distribution of these metrics in the robot's workspace is represented. Additionally, the effects of equipped end-effectors are included in the evaluation. Figure 5.1 visualises the execution sequence of the process steps to select performance metrics and calculate them in a performance map.

In the course of the chapter, the process steps are presented as depicted in figure 5.1. At first, the preselection of performance metrics to quantify a robot's capabilities to perform specific assembly tasks is presented and based on the chosen performance metrics, the process of generating a performance map is detailed in section 5.1, estimating and visualising the performance metric values in the robot's workspace. Since assembly operations require a robot arm and an end-effector to execute the task, section 5.2 includes the influences of the equipped end-effector in the executability evaluation. The results of validating and verifying the proposed methods are presented in section 5.3, and the chapter's findings are summarised and concluded in section 5.4.

During the research, chronologically, the first major design cycle focused on the induced functional redundancy through the effect of end-effector and the inclusion in the workspace evaluation, as published in [KLUG23e]. Since the provided method of including the effect of end-effectors in the first design cycle was only applied on capability maps, which is limited to measuring the reachability of a point or pose by the robot, the second major design cycle extended the method to further performance indices. In this second design cycle, a matching of performance metrics to assembly task types was introduced, enabling the calculation of a "performance map", representing the distribution of multiple robot performance metrics, as described in [KLUG24b]. The performance maps and the inclusion of end-effectors enable the evaluation of the executability of assembly tasks by robots with specific performance metrics.

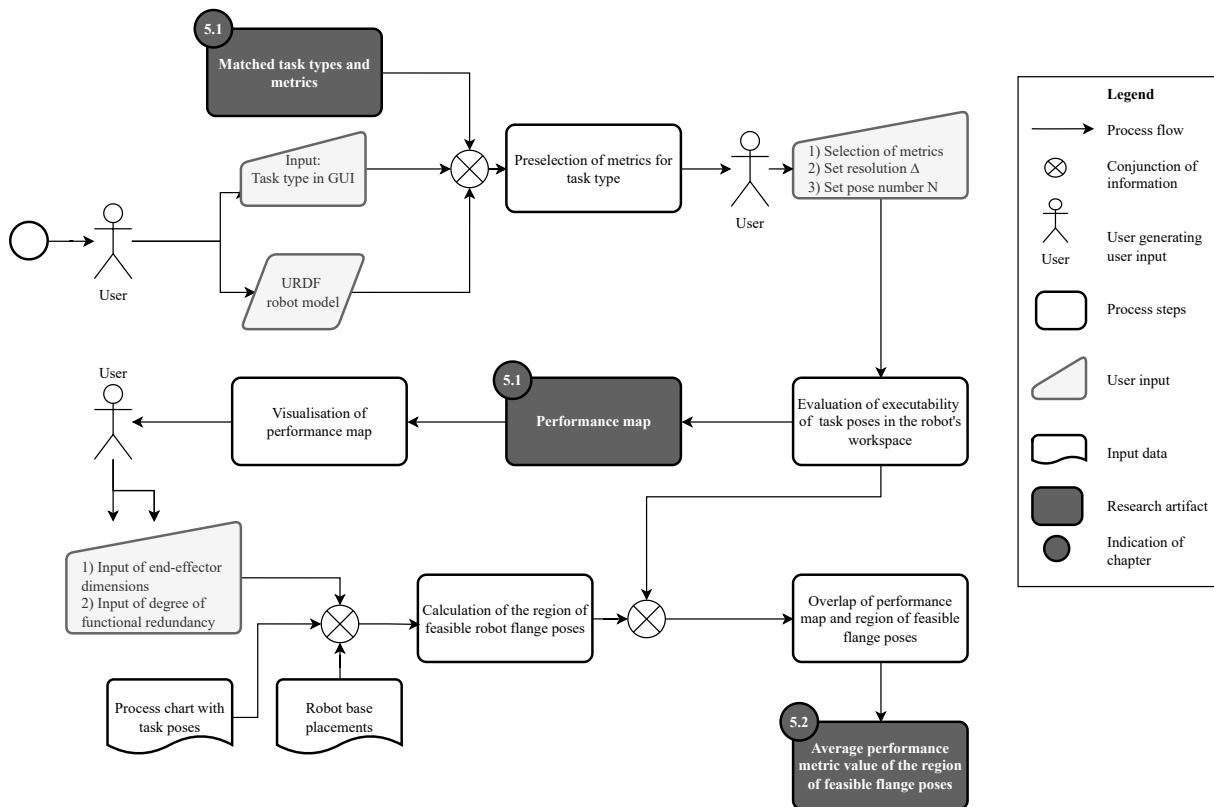


Figure 5.1: Process overview of evaluating the executability of assembly tasks by a robot — activity diagram.

Prior to the development of the executability evaluation, the requirements are derived from the research deficits outlined in section 3.4 and summarised in figure 3.3. They set the parameters and goals for investigating the second underlying research question. The first necessary condition for the executability of an assembly task by a robot is the reachability of the pose, which is dependent on the robot's spatial placement in relation to the pose. Additionally, the executability of each task type has to be evaluated based on performance metrics tailored to that task type, as different tasks require different robot performance (e.g., constant end-effector velocity or high accuracy). As end-effectors impact the size, shape, and quality of the robot's workspace, their effect has to be accounted for. Moreover, to ensure extendability to additional task types, different performance metrics must be enabled in the future. Finally, the executability distribution of different robots should be visualised to assist human operators in the planning of robot base placement. Additionally, the visualisation enables the validation technique of animation to quickly assess, whether the resulting executability representation is realistic. The intended application for performance maps in assembly formation planning poses the following **requirements**:

- RQ01 Representation of the reachability of poses in the workspace by the robot flange
- RQ02 Representation of the executability of task poses in dependence on the task type
- RQ03 Inclusion of the effect of equipped end-effectors into the workspace evaluation
- RQ04 Easy inclusion of further performance metrics for robot workspace evaluation in the future
- RQ05 Visual representation to enable readability by humans

5.1 Evaluating a robot's workspace with different performance metrics

A robot's performance is not isotropic across its workspace due to its mechanical structure. Consequently, it is paramount to determine the specific areas within the robot's workspace where it possesses the required performance metrics. The following section presents first a matching of metrics to tasks and second a novel method to calculate the selected performance metrics in the so-called performance map.

The executability of different task types depends on different parameters (e.g. constant end-effector speed for spray painting or repeatable path and pose accuracy for positioning small parts). Performance metrics evaluate a robot's capability to perform a given assembly task. Hence, a method is needed to determine suitable performance metrics for evaluating specific task types. To address this research gap, a method is introduced to match performance metrics with assembly task types.

Relevant metrics must be identified and defined to evaluate the robot's performance and suitability for the task at hand. To comprehensively capture various aspects of robot behaviour, it is necessary to use distinct and independent metrics with unique formulations since these metrics will enable a more accurate and comprehensive analysis of the robot's performance.

The main distinctions between task types during execution have been identified as the type of movement (continuous or discrete), precision (high or low), and feasible TCP range at the goal pose (reachable or dexterous) [KLUG24b]. Thereby, reachable indicates multiple feasible orientations at the task position, whilst dexterous indicates that either only one or a small number of orientations are allowed. Each category is associated with a metric chosen from table D.1 to quantify the required property. An overview of the task categories and their corresponding matching to performance metrics is presented in table 5.1, followed by a description of the metrics employed.

Table 5.1: Metrics for evaluation and metric categories.

Category	Specification	Matched metric	Task example
Movement	Continuous (C)	Manipulability measure w	Welding [DEUT22, p.13]
	Discrete (D)	Dexterity D_N , Range depends on TCP range	Screwing [DEUT22, p.13]
Precision	High (H)	Condition number > 0.8	Screwing [DEUT22, p.13]
	Low (L)	Condition number > 0.5	Transporting [VERE90, p.5]
TCP range	Reachable (R)	Lower dexterity range $D_N > 50$	Inspecting [DEUT95, p.6]
	Dexterous (DX)	Higher dexterity range $D_N > 80$	Part insertion [DEUT22, p.13]

Movement: A robot's capability to transition between configurations can be assessed by the scalar **manipulability measure**²⁷, denoted as w . Consequently, the manipulability measure is employed here to assess tasks involving continuous movement, while discrete movement tasks are adequately evaluated using the dexterity metric (explained below).

Precision: SALISBURY ET AL. (1982) introduced the concept of the **condition number**²⁷, denoted as κ , as a metric of the "kinematic isotropy" of the Jacobian matrix.

TCP range: The **dexterity index**²⁷ is denoted as D and was introduced by KUMAR AND WALDRON (1981). It calculates the average reachable orientation of a robot in a single position by setting the range of reachable angles in relation to the overall range of angles.

Previously, D_{RI} was used as a reachability metric for computing capability maps [ZACH13, p.10; MAKH18, p.139]. However, when employing D and D_{RI} for workspace evaluation, the distribution of positions where they are computed significantly influences the resulting reachability values representing the entire workspace.

The distributed reachability index D_N is proposed to overcome this over-representation of discrete positions. D_N is calculated per voxel as the percentage of reachable poses E over the total number of tested poses N . In difference to D_{RI} , D_N considers spatially distributed poses with equidistant positions in one voxel instead of considering multiple orientations for a singular position per voxel. Insight into calculating D_N is given in equation 5.1 and detailed in algorithms 3 and 4.

$$D_N = \frac{E}{N} \cdot 100 \quad (5.1)$$

Matching of performance metrics to tasks: Tasks that offer multiple execution orientations or a range of allowable orientations (indicated as "reachable") are ascribed to a lower minimum dexterity threshold. Conversely, tasks with only one permissible orientation (indicated as "dexterous") are assigned a higher threshold.

Table 5.2 provides an excerpt of categorising tasks according to their properties. As derived in section 4.3, the assembly operations noted in LOTTER ET AL. (2012), extended by tasks of [DEUT22; DEUT03; VERE90], are considered as assembly tasks. The full list of tasks and their categorisation can be found in the appendix in section E in table E.1.

In conclusion, a decision support system is proposed designed to aid in selecting performance metrics to evaluate the feasibility of specific assembly tasks. This system aims to empower individuals who may not have expertise in the field to make informed choices regarding metrics for evaluating the execution of tasks. The need to match performance metrics with specific robotic assembly tasks is addressed. By categorising tasks based on movement type, precision, and TCP range, the approach provides a clear framework for metric selection. Introducing the distributed reachability index D_N enhances evaluation, offering nuanced insights. The method advances the field by facilitating a more accurate assessment of a robot's capabilities across diverse assembly tasks.

²⁷Introduced in section 2.2.2. Further insight into the metric is provided in the appendix D.

Table 5.2: Matching of performance metrics to tasks.

Task type	Task	Movement (Discrete or Continuous)	Precision (High or Low)	TCP range (Reachable or Dexterous)
Assembling [DEUT03]	Putting together	D	H	DX
	Filling	D	L	R
	Pressing (in)	D	H	DX
	Joining by forming	C	H	R
	Welding	C	H	R
	Soldering	C	H	R
	Gluing	C	H	R
Handling [VERE90]	Storing	D	L	R
	Transporting	D	L	R
	Moving	C	L	R
	Controlling: Inspecting	C	L	R
	Controlling: Measuring	D	H	R
Separating [DEUT22]	Drill	D	H	DX
	Disassembling	C	L	R
Coating [DEUT22]	Painting, Lacquering	C	L	DX
	Printing	C	H	R
	Labelling	C	L	R
	Spatulaing	C	L	R
	Plastering	C	L	R

As detailed in section 2.2.1 analytic, geometrical and numerical approaches exist to **compute workspaces** [CAST08, p.15; ZHAN19a, p.3837; DASH05, p.777]. Since for analytical and geometric computation, solvers have to be developed for each robot [CAST08, p.15; PATE15; ZHAN19a, p.3837; VAHR15, p.785], a numeric approach is chosen here. Calculating numeric workspace maps (e.g., capability map) to aid in robot base placements reduces the computation time thousandfold over generating IK solutions for each new base placement and configuration [MALH21, p.15]. Thus, this approach is favoured here. Numerical workspace discretisation methods transform the robot's reachable workspace into discrete voxels — equidistant cubes to simplify index calculation and representation.

A computational method calculates the **performance map** for the chosen metrics. The evaluation process focuses on determining the executability of the task poses within the robot's workspace, providing valuable insights into its performance and capability. In difference to former approaches, the performance map allows for the computation of matched performance metrics. Additionally, the metric values are calculated for distributed poses in each voxel instead of only considering one position with multiple orientations per voxel. The steps for calculating the performance map will be detailed below.

As depicted in figure 5.1, the initial stages of assessing the executability of particular assembly tasks involve the generation of a performance map. The user initiates the process by providing input through the General User Interface (GUI), specifying the type of assembly task and the robot's specifications in the form of a URDF. Based on the matched metrics, the robot's capability to perform the given assembly task type are queried from a database. These preselected metrics are presented to the user. Next, the user can choose the specific metrics to be included in the evaluation from the preselected metrics, tailoring the assessment to their needs.

Defining the parameters for the performance map, the user sets the desired voxel resolution Δ and angle resolution $\frac{1}{N}$, where N represents the number of poses per voxel. The size of the performance maps voxels is determined by the resolution Δ , as it defines the side length of the voxels $(\Delta x, \Delta y, \Delta z)$, as can be seen on the left-hand side of figure 5.2. For metrics that rely on a specific orientation, the angle resolution $(\Delta\alpha, \Delta\beta, \Delta\gamma)$ can also be determined. Thus, the robot's workspace is divided into I voxels, depending on the maximum robot reach R and the resolution Δ , as indicated below in equation 5.2.

$$I = \frac{4R^3}{\Delta^3} \quad (5.2)$$

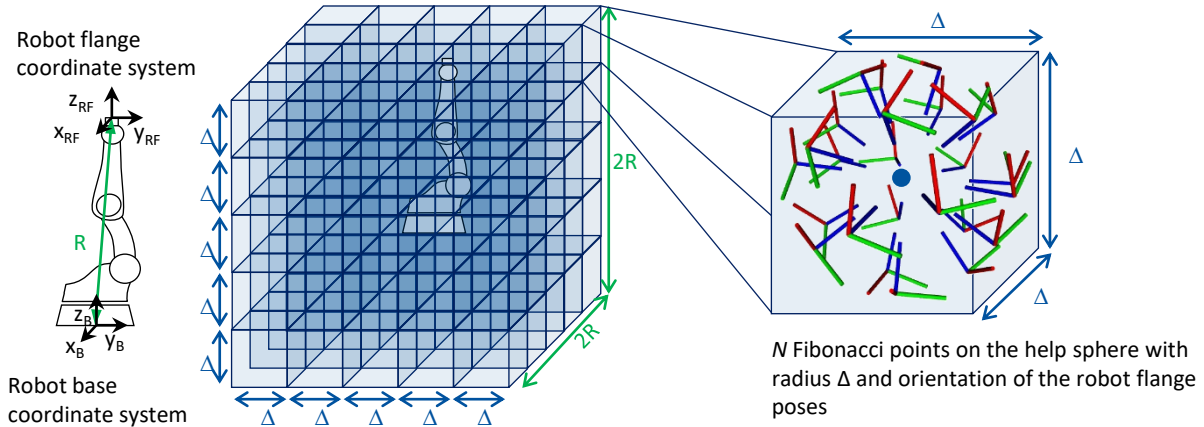


Figure 5.2: Pose calculation for performance map creation.

Left: Discretisation of the robot workspace into I voxels in dependence on the robot reach R and the resolution Δ . Right: The representation of N poses (the robot flange goal poses) per voxel is equally distributed over the help sphere in the voxel.

A discrete number of robot configurations is used to calculate the average performance metric for each voxel. The calculated metric values are mapped to discrete workspace elements, representing the robot's capability in its workspace. To generate N poses per voxel v_i with $i \in [1, I]$, to compute the discrete performance metric values. First, the centre of each voxel is identified. Secondly, a sphere with radius Δ is created with the voxel's centre as the centre. Here, the Fibonacci algorithm, as introduced in [SWIN06, p.1772], is utilised to distribute N equidistant poses across the surface of the sphere. To generate a uniform distribution over the sphere's surface, the points are distributed with increments of the golden angle ϕ . The algorithmic distribution of N Fibonacci points on the sphere surface is detailed in algorithm 3, and the creation of the pose orientation for each point is detailed in algorithm 4.

Algorithm 2 Calculating a performance map.

```

1: Input: URDF as robot model, max. reach  $R$ , resolution  $\Delta$ ,  $N$  poses per voxel  $p_{i,n}$ , chosen metrics
2: Output: Performance map
3: Compute number of voxels  $I$  according to Eq. 5.3
4:  $i \leftarrow 0$ 
5: for  $i \leq I$  do
6:    $i \leftarrow i + 1$ 
7:   Calculate poses  $p_{i,n}$  of voxel  $i$  according to Alg. 3, 4
8:    $n \leftarrow 0$ 
9:   for  $n \leq N$  do
10:     $n \leftarrow n + 1$ 
11:    if IK solution for pose  $p_{i,n}$  then
12:       $E_n \leftarrow 1$ 
13:      Calculate chosen metric for pose  $p_{i,n}$ 
14:    else if No IK Solution for pose  $p_{i,n}$  then
15:       $E_n \leftarrow 0$ 
16:    end if
17:  end for
18:  Compute  $D_N = \frac{E}{N} \cdot 100$  with  $E = \sum_{n=1}^N E_n$ 
19: end for

```

The discrete values of the chosen performance metrics are calculated for the N finite poses per voxel. Equation 5.3 indicates the resulting p distributed poses over the workspace.

$$p = N \cdot \frac{4R^3}{\Delta^3} = N \cdot I \quad (5.3)$$

With the maximum robot reach R , and I as the total number of voxels. The computed values of the performance metrics are then averaged for each voxel.

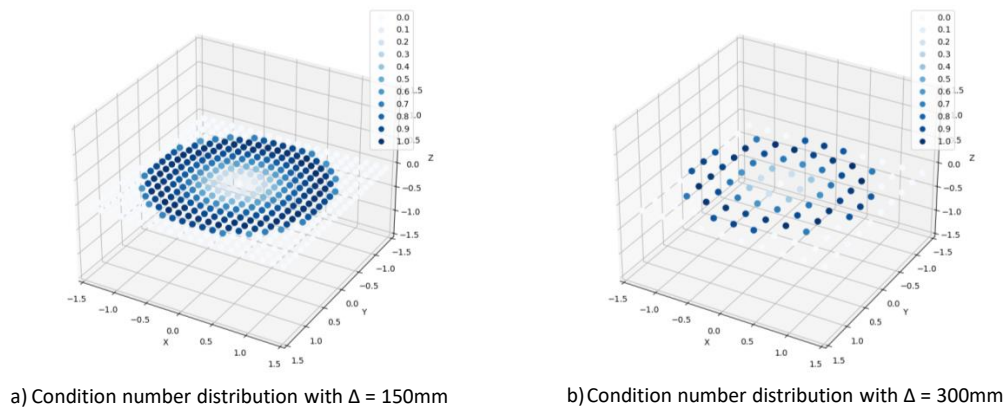


Figure 5.3: Exemplary results of the generated performance map of a UR10, with different workspace resolution Δ visualised as cross sections of the performance maps.

Finally, the results are visualised, presenting a performance map representing the robot's proficiency in handling the given assembly task. After the performance map values have been computed, they are graphically represented using Python and Matplotlib, as referenced in [CASW]. This visual representation, as depicted in figures 5.3 – 5.4, offers assembly layout

Algorithm 3 Calculating N Fibonacci points on a sphere surface.

```

1: Input: Number of poses per sphere  $N$ , Resolution  $\Delta$  as sphere radius, centre coordinates centre
2: Output: List of Fibonacci points points
3: Initialise empty list points
4:  $\phi \leftarrow \pi \cdot (3.0 - \sqrt{5.0})$  ▷ Golden angle in radians
5:  $n \leftarrow 0$ 
6: for  $n \leq N$  do
7:    $n \leftarrow n + 1$ 
8:    $y \leftarrow 1.0 - \frac{n}{N-1} \cdot 2.0$  ▷  $y$  goes from 1 to -1
9:    $\zeta \leftarrow \phi \cdot n$  ▷ Golden angle increment
10:   $x \leftarrow \cos(\zeta) \cdot \text{radius} \cdot \sqrt{1.0 - y^2} + \text{centre}[0]$ 
11:   $y \leftarrow y \cdot \text{radius} + \text{centre}[1]$ 
12:   $z \leftarrow \sin(\zeta) \cdot \text{radius} \cdot \sqrt{1.0 - y^2} + \text{centre}[2]$ 
13:  Append  $(x, y, z)$  to points
14: end for

```

Algorithm 4 Calculate orientation of the N Fibonacci points on the sphere surface.

```

1: Input: Centre coordinates centre, List of  $N$  Fibonacci points points, Resolution  $\Delta$ ,
2: Output: List of  $N$  poses poses
3:  $n \leftarrow 0$ 
4: for  $n \leq N$  do
5:   Get  $\text{point}_n$  from points
6:    $v_{CP} \leftarrow (\text{point}_n - \text{centre})$  ▷ Vector from centre to  $\text{point}_n$ 
7:    $\zeta \leftarrow \arccos\left(\frac{v_{CP}[2]}{\Delta}\right)$  ▷ Inclination angle
8:    $\phi \leftarrow \arctan(v_{CP}[1], v_{CP}[0])$  ▷ Azimuthal angle
9:    $\text{roll} \leftarrow 0$  ▷ Set roll angle
10:   $\text{pitch} \leftarrow \pi + \zeta$  ▷ Set pitch angle
11:   $\text{yaw} \leftarrow \phi$  ▷ Set yaw angle
12:  Append  $(x, y, z, \text{roll}, \text{pitch}, \text{yaw})$  to poses
13: end for

```

planners and engineers an intuitive means to understand the distribution of local performance metrics in the robot workspace.

The generated performance map can be extended to include the effects of end-effectors, as presented in section 5.2 or for a feasible base placement of the robot to execute the assembly tasks, as presented in chapter 6.

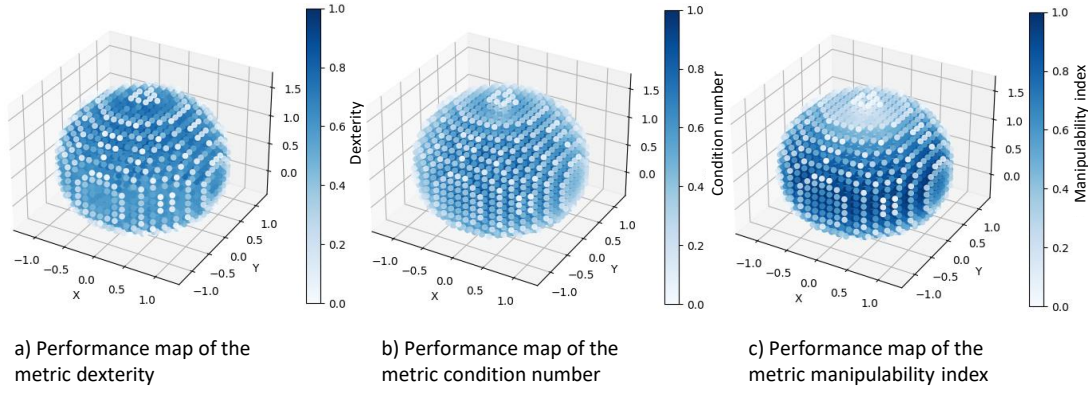


Figure 5.4: Exemplary results of the generated performance map of an ABB IRB1600-1.2 with a workspace resolution Δ of 120mm, and 25 poses per voxel.

5.2 End-effector-dependent executability evaluation

While existing research (reviewed in [PATE15; ANGE16; RUSS22], and detailed in section 2.2) defines metrics for evaluating robot characteristics or focuses on computing workspaces with these metrics, the impact of end-effectors on the robot's workspace and reachability is neglected. Executing assembly tasks requires not only a robot arm but also an equipped end-effector. Thus, the need arises to include the effects of end-effectors for workspace evaluation. Not only does the equipped end-effector shift the point of interest from the robot flange to the TCP, but depending on the considered end-effector and the allocated assembly tasks, functional redundancy²⁸ occurs.

The following chapter details a procedure to include different end-effectors and their effects in the workspace evaluation. Depending on the equipped end-effector's Degrees of Freedom (DoFs) and dimension, a single task pose is expanded into a region of feasible task poses. To evaluate the task's executability, the region of feasible task poses is overlapped with the capability map or performance map, and the average of the relevant metrics is calculated. If the mean value of the metric surpasses a predetermined threshold, it suggests a high likelihood that the task is executable.

In the first design cycle, the proposed identification of feasible task poses dependent on the end-effector was validated with capability maps. The results were published in [KLUG23d; KLUG23e]. The second design cycle transfers the methodology to the performance maps, incorporating multiple performance metrics.

The proposed process for evaluating the end-effector-dependent executability for a specific end-effector and robot is structured into the following steps:

1. The user specifies the dimensions (length, height, width (x_{EE}, y_{EE}, z_{EE})) and degree of functional redundancy of the combination of task type and end-effector.
2. The positions of the robot base placements are retrieved (will be detailed in chapter 6), and the TCP task poses are taken from the process chart.

²⁸Functional redundancies occur when the system is indeterminate.

3. The region of feasible Robot Flange (RF) poses is calculated in relation to the given TCP task pose and depending on the end-effector.
4. Either the capability map or performance map is generated (or is called upon if already computed).
5. The region of feasible robot flange poses is overlapped with the capability map or performance map at the robot's base placement, and the metric value for each flange pose is determined.
6. The average performance metric value for the region of feasible robot flange poses is calculated.

Figure 5.1 integrates the process steps into the entire process of evaluating the executability of assembly tasks by a robot, whilst figure 5.5 highlights the process of the end-effector-dependent executability evaluation. A GUI was created to specify the input parameters, as shown in figure 5.5 in the upper left. The GUI allows manual input of the end-effector parameters and the robot's capability map or performance map. The calculation operates in the background, providing the user with visual representations of evaluated workspaces and quantitative metrics.

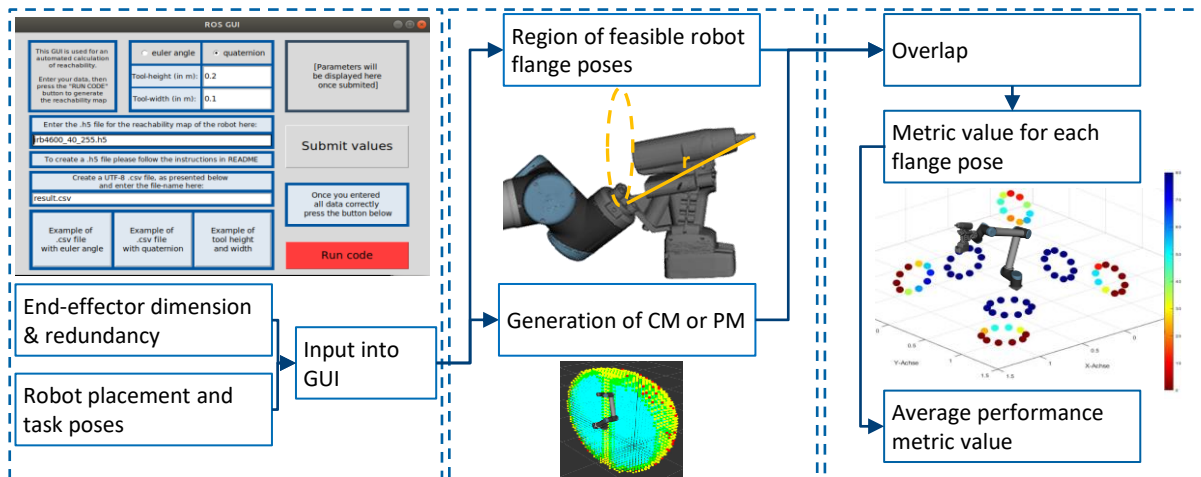


Figure 5.5: Process steps for evaluating the end-effector-dependent executability by estimating a region of feasible robot flange poses and the average performance metric value for this region.

5.2.1 Calculating the region of feasible robot flange poses

Including end-effectors in evaluating the robot workspace requires knowledge about the effect of the end-effector. On the one hand, including an end-effector shifts the point of interest from the robot flange to the TCP, as can be seen in figure 5.6 a) highlighted in yellow as radius r . This shift can be expressed by translation and rotation depending on the dimensions of the end-effector. On the other hand, executing a task with an equipped end-effector can create functional redundancy and, therefore, allow for a region of feasible robot flange poses. The functional redundancy defines the form of the region of feasible robot flange poses, whilst the end-effector-dimensions define its size.

The **degree of functional redundancy** varies in dependence on task type and end-effector. The degree of functional redundancy reflects in which region the robot flange can be positioned

whilst the TCP reaches the task pose. For example, screwing a screw provides one degree of functional redundancy, as a screwdriver can rotate around its direction of impact during task execution (see figure 5.6 b)). Consequently, the region of feasible robot flange poses forms a circle (the yellow dotted circle in figure 5.6 b)). Thereby, 5.6 a) visualises the shift of the point of interest from the robot flange to TCP and end-effector dimension x_{EE}, y_{EE}, z_{EE} determining the radius r to calculate the possible robot flange poses.

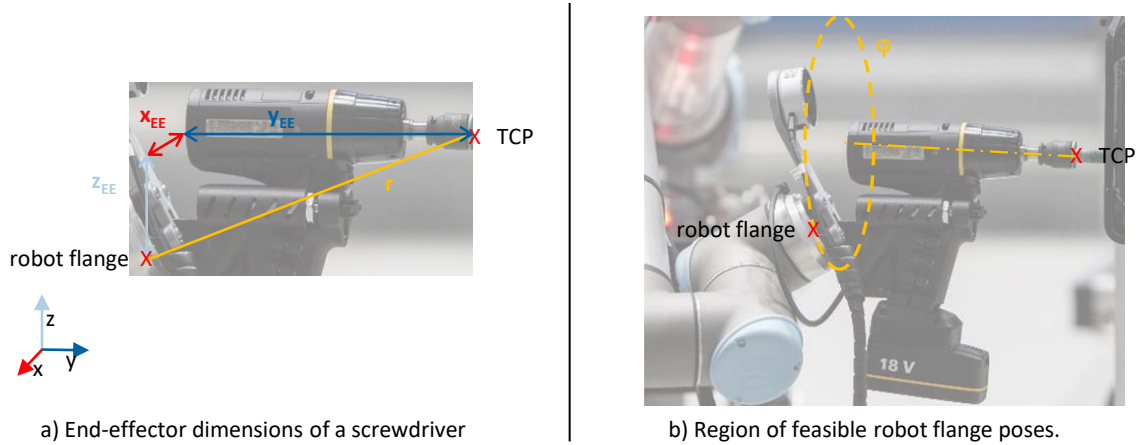


Figure 5.6: Determination of the feasible robot flange region in dependence on the end-effector dimensions, exemplified by a screwdriver as end-effector.

Figure 5.7 provides an example of gripping a sphere with a parallel jaw gripper. If the absence of obstacles in close proximity to both the robot and the sphere is assumed and self-collision is neglected, the robot's flange could rotate up to a full 360° degrees around Φ, θ and φ of the TCP (see figure 5.7 b)). Therefore, gripping a sphere provides three degrees of functional redundancy, which creates a sphere as the region of feasible robot flange poses (cf. figure 5.7 b)–c)). In figure 5.7 c)–d) alternative gripping poses are depicted.

Figure 5.8 visualises two additional examples of gripping an object. On the left-hand side (figure 5.8 a)), a parallel gripper gripping a cuboid object with alternative gripping poses is visualised. As the cuboid can only be gripped from two opposing sides, one degree of functional redundancy results from rotating the gripper around its x-axis. If the parallel jaw gripper handles a cylindrical object (figure 5.8 b)), the mantle surface of the cylinder can be used as the gripping surface. For a successful grip, the x-axis of the gripper must be positioned perpendicular to the mantle surface. Due to the rational, symmetrical shape of the cylinder, the first degree of functional redundancy results from the rotation around the y-axis and the second from a rotation around the x-axis.

Following the provided train of thought, all combinations of end-effectors with assembly tasks can be categorised according to the resulting degrees of functional redundancy. Here, only rotational and not translational degrees of functional redundancy are considered. Additionally, as the purpose of the determination of the region of feasible robot flange poses is to overlap the region with the capability map or performance map to estimate the performance metric values for this region, only a representation of either the circle or sphere surface is required. If the robot's joints are limited or obstacles in the vicinity restrict the robot's motion, the attainable rotation angles become constrained. Consequently, only segments of the circle or

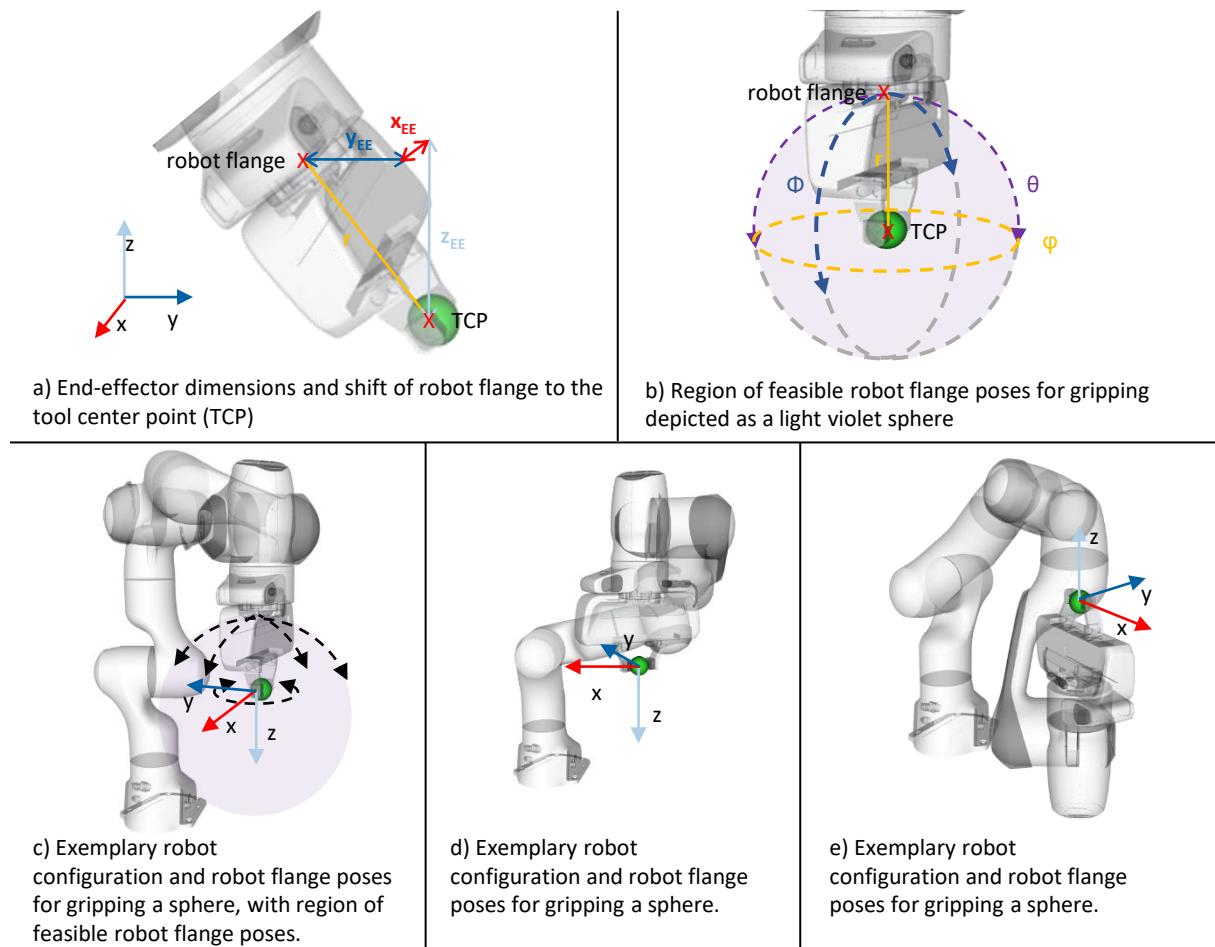


Figure 5.7: Region of feasible robot flange poses for a gripping task — depicted as a light violet sphere, in dependence on the end-effector dimensions and shift of robot flange to TCP.

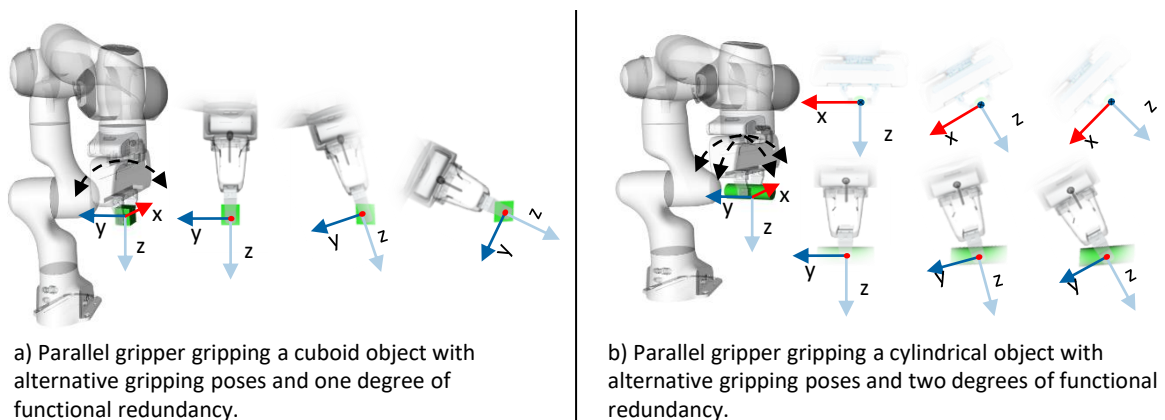


Figure 5.8: Two examples of a parallel gripper gripping objects with alternative gripping poses and varying degrees of functional redundancy.

sphere surface remain as the feasible region for robot flange poses. Taking the rotational degrees of functional redundancy and range of rotation angles as input, one can determine the two-dimensional region of feasible task poses in the three-dimensional space $\mathbb{R}^3$. The radius r depends on the end-effector's dimensions, as visualised in figure 5.7 and 5.6 and calculated according to equation 5.4.

$$r = \sqrt{x_{EE}^2 + y_{EE}^2 + z_{EE}^2} \quad (5.4)$$

First, the centre $(x_0, y_0, z_0)^T$ of the region of feasible robot flange poses is determined in equation 5.5. The centre depends on the end-effector's dimensions x_{EE}, y_{EE}, z_{EE} and the goal pose of the TCP x_1, y_1, z_1 .

$$(x_0, y_0, z_0)^T = (x_1, y_1, z_1)^T + (x_{EE}, y_{EE}, z_{EE})^T \quad (5.5)$$

The points in the region of feasible robot flange poses follow as derived in equations 5.6–5.9.

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2. \quad (5.6)$$

with

$$x = r \sin \theta \cos \varphi \quad (5.7)$$

$$y = r \sin \theta \sin \varphi \quad (5.8)$$

$$z = r \cos \theta \quad (5.9)$$

and θ and φ as reachable angles of the rotational degrees of redundancy of the end-effector and $0^\circ \leq \theta \leq 180^\circ$, $0^\circ \leq \varphi < 360^\circ$ and $r \geq 0$. When the end-effector provides one degree of functional redundancy, this results in a circle, and for more than one degree of functional redundancy, it results in a spherical surface or, if limitations occur, segments thereof. The radius of this area is determined by the distance between the TCP and the flange and is represented as " r ".

The region of feasible robot flange poses depicts all possible poses of the robot that ensure the correct pose of the end-effector in relation to the task pose. It is calculated in reference to the task pose, the end-effector dimensions and the degree of redundancy. Here, all assembly tasks are assumed to be represented as 6D goal poses or trajectories, which can be discretised into a sequence of 6D goal poses.

In difference to [MALH21], not only end-effectors with one or two DoFs can be considered, and the method of including the end-effectors effect does not require manual simulation of every goal pose. Therefore, the proposed method provides a higher degree of automation.

5.2.2 Determining the average performance metric value for the region of feasible robot flange poses

Following the computation of the region encapsulating feasible robot flange poses, the next step is to evaluate the reachability and compliance with the assigned metric limits. Consequently, the computation of the average performance metrics — indicating the executability — for the

region of feasible poses is imperative. A higher average performance metric within the region indicates a heightened probability that the robot can effectively execute the assembly task.

To determine the average performance metric, the resulting region is overlapped with either the capability map — indicating the likelihood of reachability — or the performance map — indicating executability. Therefore, the values of the performance metrics of the voxels of either the capability map or the performance map closest to the region are taken to evaluate executability. As a result, the robot flange poses are assigned the performance metric values of the voxels with the smallest Euclidean distance between the robot flange pose and the voxel's centre. The performance metrics average p_{ma} is given in equation 5.10. The p_{ma} is derived by dividing the sum of all performance metric values p_{mv} per region through the number of tested robot flange poses p .

$$p_{ma} = \frac{1}{p} \sum_{i=1}^p p_{mv,i} \quad (5.10)$$

5.3 Verification and validation of the workspace analysis

The artifact of executability evaluation has to be verified and validated [HEVN07]. The central component of the validation is the examination of the operational validity. To test the practical applicability of the developed artifacts, they are tested and applied in their application domain. The following section summarises the results of this evaluation and validation. The underlying methodology and applied techniques are explained in section 2.5.

The validity of the proposed executability evaluation is presented by examining the degrees of the fulfilment of the requirements stated at the beginning of the chapter. The goal is to represent reachability and executability while including the effect of end-effectors. For further use, the inclusion of additional metrics has to be enabled. Additionally, reachability and executability should be represented visually to enable human readability.

Conceptual model validation

A novel evaluation method is introduced, encompassing the impact of end-effectors on estimating robot performance. Validating the hypothesis that the performance metrics average p_{ma} correlates with the analytic performance metric value requires testing for different poses and comparing the results.

Figure 5.9 visualises the average reachability index's calculated values in comparison with the solution of inverse kinematics. The reachability index is achieved by overlapping the robot's performance map for reachability values with the region of feasible robot flange poses. As the visualisation shows, regions with a low reachability index (orange and red in figure 5.9 a)) correlate to areas where no inverse kinematics solutions can be found (dark red in figure 5.9 b)). This first validation by animation indicates a correlation, which has to be further validated in the following.

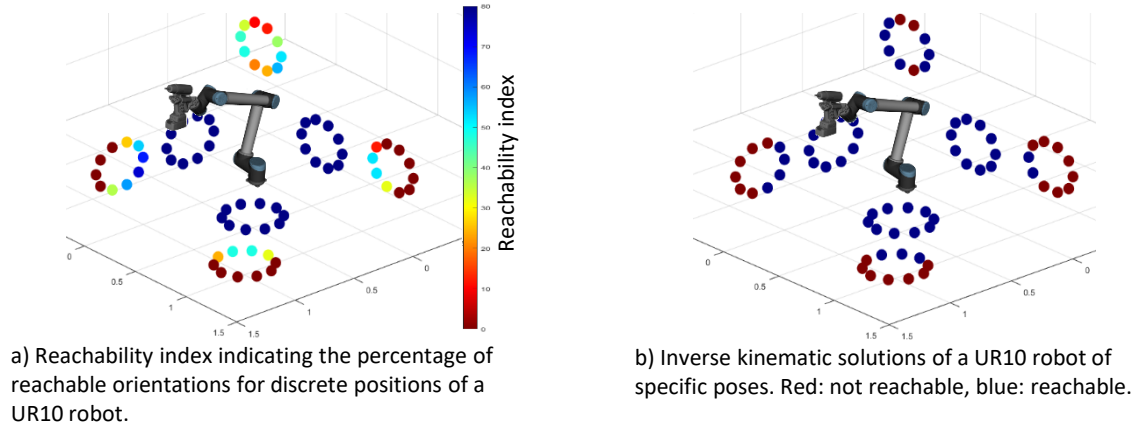


Figure 5.9: Juxtaposition of the average reachability indexes calculated values and the solution of inverse kinematics for the discretised region of feasible poses for screwing tasks distributed in the robot workspace of a UR10 robot.

The second testing is based on figure 5.9. A screwdriver with the dimensions of 0.2 m height (z_{EE}) and 0.1 m length (y_{EE}) is chosen. Secondly, the region of feasible robot flange poses according to equation 5.6 is calculated. A capability map with 0.05 m resolution is taken to represent the robot's reachability. The test was executed on Intel Core i5-4210M CPU 2.60 GHz x 4 with Ubuntu 18.04 and ROS Melodic. 39 end-effector task poses in the robot workspace were taken. With a discretisation of 10 task poses per region of feasible robot flange poses, 390 poses were tested for reachability. The results are shown in figure 5.10 a).

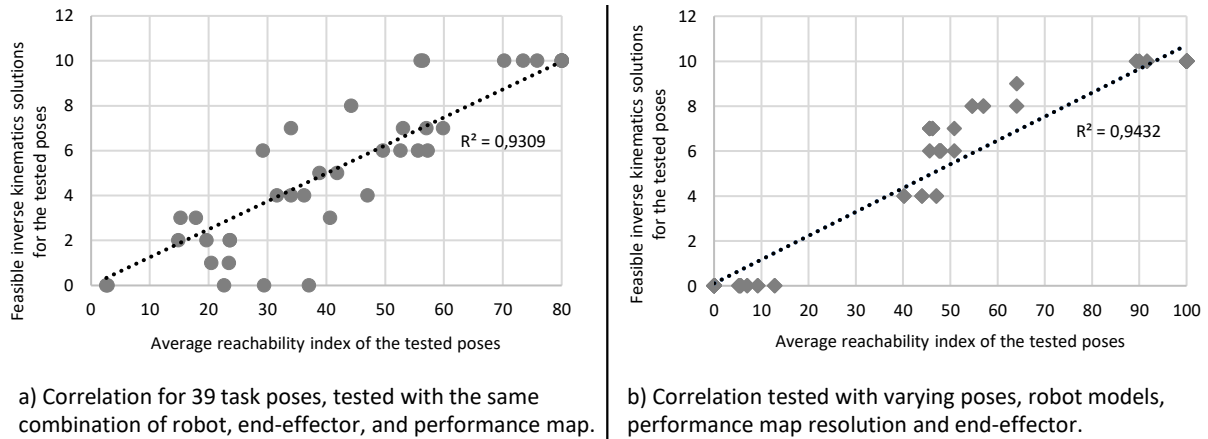


Figure 5.10: Correlation of the average reachability index as performance metrics average p_{ma} and the inverse kinematics solutions of the same task poses.

In a third experiment, the influence of different variables is tested. The following influence variables are identified for the proposed method of end-effector-dependent feasible task poses: robot model, task type, task pose, end-effector-type, and resolution Δ of the performance map or capability map. Thereby, the end-effector type influences the degree of functional redundancy, and the size of the region of feasible robot flange poses depends on the end-effector dimensions (length, height, width).

The validation procedure for assessing the executability of end-effector-dependent processes is demonstrated through the example of the screw-driving operation. Hence, an electric screwdriver serves as the end-effector, and a UR10 robot arm is utilised, as illustrated in figure 5.6. For screwing a screw, the end-effector can be placed anywhere on the yellow dotted line in figure 5.6 a), derived by rotating the end-effector around its z-axis. Thus, per the suggested procedure, the first step involves setting the functional redundancy of the end-effector to one. Investigating the influence of the five predictor variables task pose, end-effector dimension, robot model, performance map resolution, and the utilised IKsolver, a Design of Experiment (DoE) is conducted. Three representative task poses distributed across the workspace are selected. Two end-effector dimensions (assuming colinearity of the TCP and end-effector z-axes) are considered $[(z_{EE}, y_{EE})$ as (0.1 m, 0.2 m) and (0.2 m, 0.1 m)]. The robot models UR10 and UR5 are utilised, and three resolutions Δ of the performance map (0.05 m, 0.08 m and 0.1 m) are set. For the IK solver TRAC_IK and KDL are employed [SOET20; BEES15; BEES21].

Following a full factorial design, 72 experiments are conducted²⁹. These tests are randomly organised as a single experimental block. The calculated performance metric value and the performance metrics average p_{ma} are compared with the exact value per pose for all 72 experiments. The resulting correlation is visualised in figure 5.10 b).

A correlation is observed between a higher number of reachable robot flange poses (solved by IK) and a greater likelihood of executability when a high p_{ma} value is identified. In regions characterised by a high p_{ma} value, the possibility of pose non-reachability persists, while in areas where $D_{RI} = 0$, it is consistently true that the associated position is beyond the operational range of the manipulator.

Computerised model verification

The verification of the computerised model ensures the accurate implementation of the conceptual model. To verify these functionalities in the computerised model, animation techniques are employed to observe the discretisation and visualisation of the performance map, taking into account end-effector effects. Animation proves to be particularly beneficial as it facilitates intuitive error interpretation. Furthermore, the objective includes offering a human-readable interpretation of robot workspaces, and animation has demonstrated its utility in illustrating areas where the robot can or cannot perform assembly tasks.

Operational validation

The generation method was tested to validate that the performance map represents the spatial distribution of performance metrics. Generally, this method should apply to any PUMA robot for which a URDF file is available. As in the previous tests, the robot models of a UR5 and a UR10 robot were taken and the performance maps were created, here, it is tested on the

²⁹The conducted tests, including the input data, the listing of the full factorial DoE as well as the results are published under <https://doi.org/10.5281/zenodo.10119467> [KLUG23c]

URDF files of ABB IRB120-0.58 and ABB IRB1600-1.2 robots, to validate applicability to different robot types and URDFs.

Investigating the influence of three predictor variables on the robot manipulation performance metrics, a DoE is conducted. The predictor variables considered are tested with respective minimum and maximum values: the resolution (Δ) with a range of 0.08 m to 0.12 m, the number of poses per voxel (N) with options of 25 and 50 poses, and the two aforementioned robot models. Eight experiments were conducted following a full factorial design, as detailed in table 5.3³⁰.

Table 5.3: Design of experiment for conceptual model validation of the performance map.

Run order	Model	Δ Resolution [m]	N Poses per voxel
1	IRB1600-1.2	0.08	25
2	IRB120-0.58	0.12	25
3	IRB120-0.58	0.12	50
4	IRB120-0.58	0.08	25
5	IRB1600-1.2	0.08	50
6	IRB1600-1.2	0.12	50
7	IRB120-0.58	0.08	50
8	IRB1600-1.2	0.12	25

In each experiment, the method generated the corresponding number of poses denoted as p , and the IK solver known as the "Kinematics and Dynamics Library (KDL)," provided by the Orocos Project [SOET20], was employed to process each pose. Utilising the Jacobian matrix generated in Movelt [COLE14], calculations for the condition number and manipulability measure were executed as previously described. In addition, the computation of the distributed reachability index, referred to as " D_N " (see section 5.1), was performed following the procedure outlined in algorithm 5.1. The resulting performance maps are illustrated in the appendix in section E in figures E.1, E.2 and E.3.

Changing the resolution Δ resulted in an average change of $\pm 2\%$ of the metric values. Variation of the pose number N resulted in -3% for the lower amount of poses.

Data validity

Generating the performance map relies on the robot URDF model and manual input of task type, the desired performance metrics, the resolution Δ and the pose number N . The end-effector-dependent calculation of executability relies on the performance map as an input, the manual input of the end-effector dimensions and the degree of functional redundancy. Additionally, the task poses in relation to the robot base have to be given.

Observed errors occurred mainly due to faulty human input, e.g., when a larger or smaller robot reach was given during the computation of the performance map. Thus, the resulting visualisation either shows a filled square visualisation area without the typical sphere-like shape

³⁰The conducted tests, including the input data, the listing of the full factorial DoE as well as the results are published under <https://doi.org/10.5281/zenodo.10122242> [KLUG24a].

of the robot workspace. If the reach is chosen too high, the workspace's representation appears very small compared to the visualised area.

The manual and visual interpretation of the visualised performance map or region of feasible robot flange poses requires some prior knowledge of the operator of the expected outcome. Therefore, there is no guarantee that every input error will be detected through human monitoring in the case of new input or untested scenarios.

5.4 Interim conclusion: Executability evaluation

In this chapter, the evaluation of the executability of assembly tasks by mobile robots is conducted to address the second underlying research question URQ2: ***Can the executability evaluation of assembly tasks be extended whilst considering equipped end-effectors on mobile robots?*** Performance metrics representing a robot's capability to execute tasks are calculated. The distribution of the metrics is represented and can be visualised in a performance map, depicting the valid workspace of the mobile robot. The performance metrics are chosen to evaluate the robot's capabilities for typical assembly tasks. The effects of equipped end-effectors are included in the evaluation, a facet previously overlooked in workspace evaluations within the existing literature. Therefore, the underlying research question URQ2 can be positively answered. The contribution of the chapter is summarised in figure 5.11.

The process of planning suitable formations of mobile assembly robots requires performing a workspace analysis to evaluate the reachability and executability of assembly task poses in the robot's workspace. This chapter contributes a **matching of typical assembly tasks with metrics** designed to evaluate the capability of a robot to execute this specific type of task. The **performance map** serves as a discretised representation of metric values distributed across the robot's workspace, approximating the volume of the valid workspace that aligns with the desired robot capabilities. The visualisation of this map of robot performance metrics allows for a manual examination of the robot's workspace, facilitating the identification of areas where assembly tasks may not be executed with a sufficient level of executability. By evaluating which regions the robot can access while adhering to the specified metric thresholds, one can determine the tasks that the robot can proficiently execute.

A method is presented to consider the **impact of an equipped end-effector on the workspace analysis of a robot**. The determination of **feasible robot flange poses** for task execution is contingent upon the type and dimensions of the equipped end-effector and the degree of functional redundancy derived from the combination of the end-effector with the task type. The quantitative **average performance metric** p_{ma} is introduced, which assesses the likelihood of successfully executing an assembly task. p_{ma} can be employed to identify regions within the workspace suitable for task execution, offering a quantitative measure to optimise the placement of robots concerning specific assembly task poses. It was demonstrated that p_{ma} correlates with task executability. Furthermore, a GUI was integrated to enhance the user-friendliness of the proposed inclusion of end-effectors.

The derived research artifacts are added to the knowledge base as prescriptive knowledge in accordance to [IIVA07, p.8]. The research process and results are published in [KLUG23e; KLUG24b]. Additionally, the source code, as well as data, is publicly available³¹.

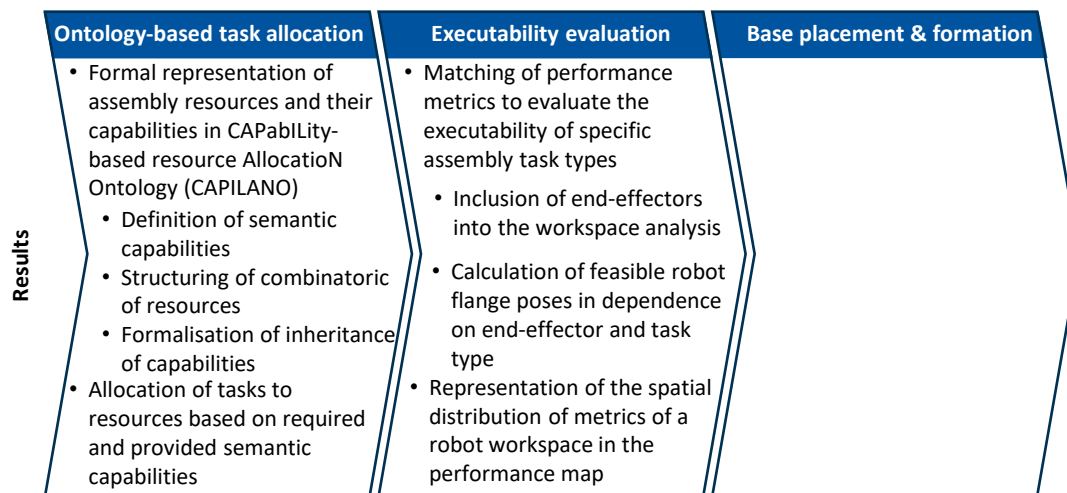


Figure 5.11: Contribution of chapter 5 to the research progress.

Critical reflection

The current discretisation and visualisation assessing the robot workspace regarding different characteristics needed to execute an assembly task does not include workspace limitations accounting for collision avoidance (neither with obstacles nor with the robot and its end-effector). Additional limitations of the reachable or executable workspace induced by, e.g. external cables connected to the end-effectors might limit the workspace further.

The functional redundancy caused by combining end-effectors and the task type is currently included manually. Moreover, the functional redundancy is currently limited to rotational redundancy and does not consider translational redundancy.

The employed task categories and the metrics used for task evaluation can be expanded by incorporating additional metrics along with explanations, thereby making the concept behind the metrics more understandable for individuals who are not experts in robot workspace evaluation and metrics.

³¹https://git-ce.rwth-aachen.de/wz1-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/end-effector-dependent-executability <https://doi.org/10.5281/zenodo.10122242>
<https://doi.org/10.5281/zenodo.1011946> [KLUG23b; KLUG23c; KLUG24a]

6 Autonomous clustering-based task allocation and robot base placement

Formation planning is a crucial component of assembly systems planning [SUEM16, p.396]. The reciprocal influence between multi-robot station formation and executability evaluation of the allocated tasks is significant, as the decoupling of these activities may yield formations in which the tasks are not executable, necessitating alterations to the station formation. Task allocation, robot workspace evaluation, and robot base placement are relevant steps to plan a feasible station formation. Simultaneous planning for robot placement and task allocation is desirable in scenarios where relocating the robot is costly or when it is undesirable to perform simultaneous base repositioning and end-effector motion [PAUS17, p.6285]. As capability-driven task allocation and end-effector-inclusive workspace evaluation are investigated in chapters 4 and 5, the consolidation of the previous considerations into a formation and base placement is pending.

This chapter treats the third underlying research question URQ3: ***How can assembly tasks be clustered to plan formations with reduced base placements of mobile assembly robots?*** As established in chapter 5, a robot's performance varies in its workspace, complicating the identification of suitable base placements. This complexity underscores the necessity to devise an efficient method for optimising the base placements of mobile manipulators. The deficit in the absence of a method for simultaneous placement of multiple mobile robots in assembly to ensure the executability of distributed tasks is answered. In this chapter, a method is presented that evaluates the task executability of formations whilst automating task allocation and simultaneous base placement of mobile robots. The research results of answering this third research guiding question are partially presented in [KLUG23a; KLUG22d].

During chronological development, the first major design cycle proposed a K-Means task clustering approach for determining robot base placements for each task cluster. K-Means proved to be suitable for clustering densely placed and continuous tasks but was not suitable for fairly distributed discrete task poses. Therefore, the additional clustering method OPTICS was added to cluster discrete tasks as part of a second major design cycle. In the third major design cycle, the method is improved to include the evaluation of the individual solutions (found clusters and base placements per cluster) depending on the metrics selected for the task at hand, as proposed in chapter 5 in section 5.1.

In the course of this chapter, first, the goal is postulated, the requirements to fulfil the postulated goal are formulated, and preconditions for the following considerations are defined in section 6.1. Section 6.2 motivates the utilisation of clustering for the goal of autonomous task allocation under spatial constraints and base placement. In section 6.3, the steps of the autonomous clustering-based task allocation and robot base placement are introduced and subsequently detailed. Following the verification and validation of the proposed method in section 6.4, the chapter concludes with a summary of the main findings in section 6.5.

6.1 Requirements and preconditions

The goal of applying task clustering algorithms is to reduce the complexity of formation planning by clustering assembly tasks into sub-task sets that are reachable from a single robot base placement. Formalised, the goal is to allocate n number of task poses to r robots placed at b base placements. The base placements are chosen within a discretised *search space* surrounding the product on which the assembly operation should be performed.

Following the requirements outlined in sections 3.4 and 3.2, discussions with industry experts and researchers as part of the relevance cycle, **requirements** are postulated for the autonomous clustering-based task allocation and robot base placement. The requirements are outlined and categorised into hard and soft requirements. Meeting the hard requirements is essential to responding positively to the underlying research question while satisfying the soft requirements would improve the outcome. If the requirements have been applied for base placement previously, a source is given.

Hard Requirements (HR)

- HR01 All task poses have to be allocated to a cluster.
- HR02 For each cluster, a valid base placement has to be found.
- HR03 A base placement is valid if all allocated task poses of the cluster are reachable by the robot placed at the base placement³².
- HR04 All task poses have to be reachable from at least one base placement.
- HR05 Collisions between the executing robot and the product must be avoided.
- HR06 Executability has to be ensured.

Soft Requirements (SR)

- SR01 The number of clusters K / base placements should be minimised.
- SR02 Each task pose should only be allocated to one cluster.
- SR03 Possibilities for collisions of multiple robots should be reduced.
- SR04 Singularities should be avoided.
- SR05 Consideration of feasible trajectories between task poses.

The following discussions on formation planning through clustering are based on the **assumptions** postulated below as a precondition for its functionality. Firstly, one condition for the applicability is solvability: At least one robot can execute the given task pose from at least one placement. Thus, for each task pose, a combination of robot arm, mobile base, end-effector, and base placement exists, which renders the task executable. This contains considerations such as ensuring that the height of the highest task pose is lower than the maximum reach of the robots. Secondly, it is assumed that the task poses remain fixed in relation to the product and world coordinate system and are not changed during the assembly process. A reduced number of base placements is assumed to be beneficial to reduce the movement of the mobile

³²Since one robot base placement must be identified for each cluster, the quantity of base placements and clusters must match.

robot bases during or between task execution. Furthermore, the objects considered in the assembly stations are limited to the product, the parts, and the robots. Thus, no additional collision avoidance (e.g. with humans in the station) is considered during task clustering. The local collision avoidance has to be taken care of during task execution and is excluded here. Scheduling and temporal optimisation of the order of executing the clusters are out of scope. Therefore, no path planning, trajectory planning or dynamic collision avoidance is carried out.

While the previous chapter 5 discussed in detail the impact of task types on executability, for the purpose of clustering, only the differentiation of continuous or discrete task poses is relevant, as it impacts the choice of the clustering method. The task types (continuous vs. discrete) are known and given before the clustering. The evaluation of task executability, as presented in chapter 5, is utilised to evaluate the feasibility of a found clustering. During the first design cycle, it is assumed that a task pose is executable if the robot can reach it. The second design cycle considers additional criteria and metrics.

6.2 Motivating and choosing clustering algorithms for task allocation and robot base placement

As introduced in section 2.4, clustering is a procedure for discovering similarity structures in (usually relatively large) data sets [ANKE99, p.49]. Although task clustering has been studied extensively in the context of search and rescue operations and scheduling and sequencing problems [JANA17; EZUG22, p.32], its potential application for base placement has not been explored [ELAN11; KLUG22d]. This section analyses the suitability of clustering algorithms to cluster assembly tasks for robot base placement. It is followed by a summary of the working principles of autonomous clustering-based task allocation and robot base placement.

For clustering assembly tasks to be assigned to mobile robots, only methods belonging to the partitional, density-based algorithms are investigated for suitability (the respective classification of clustering algorithms is introduced in section 2.4). As each task pose should only be assigned to one robot, only hard/crisp clustering methods are considered. Of the hard/crisp clustering methods, graph-based and subspace clustering are excluded as task poses are neither graph-related nor exist in different unknown sub spaces³³. As model-based clustering algorithms rely on an underlying model, which is not valid here, they are not further considered. Density-based and square error clustering methods are further considered, as they provide clusters based on the spatial closeness of the data points and tend to derive spherical clusters.

Due to its inability to identify clusters, which are not well separated [SISO12, p.84–85] CLARANS is not suitable for clustering assembly tasks for base placement.

Since its tendency to identify clusters only in dependence on density and rejection of points that are not in dense areas, DBSCAN is not suitable to cluster points for base placement since every data point to be considered corresponds to an assembly task, which has to be reached

³³Different task types or products could be seen as differentiating subspaces, but as these properties are previously known from chapters 4,5, here only data sets consisting of task poses of one task type and belonging to one product are considered.

and cannot be discarded. Additionally, the reachability of all points in one cluster by one robot (as in maximum distance from the points to each other) cannot be inherently guaranteed since DBSCAN forms arbitrary shapes of clusters.

K-Means is specialised in identifying clusters of a circular shape. This property is advantageous in robotics, as a robot's workspace can be approximated as circular, at least in the initial stages of analysis [KLUG22d]. Thus, K-Means exhibits favourable characteristics for clustering task poses within an assembly station. Given the objective of clustering task poses so that each cluster is reachable and executable by one robot from a designated base placement, K-Means' tendency to create spherical clusters around a centroid is beneficial. However, due to its limitations in handling outliers and widely distributed data points, K-Means may not be suitable for clustering tasks possessing such characteristics. Consequently, OPTICS is selected as a secondary clustering algorithm for clustering widely distributed task poses.

6.3 Concept and working principle of autonomous task clustering and base placement

The following explains the steps for the task clustering and base placement, incorporating the chosen clustering algorithms. Firstly, an overview of the steps is given, followed by a more detailed description of the respective steps.

Figure 6.1 depicts the steps determining a feasible formation of robots. The approach subsequently limits the amount of favoured base placement in the assembly station by excluding unfavourable areas one after the other and evaluating the solutions found to derive an optimised formation.

Initially, three input files are needed: the task poses as a .csv file, the robot specifications as a URDF file and the CAD model of the product. Subsequently, the clustering algorithm, K-Means or OPTICS, is applied to the task poses based on the nature of the task — continuous or discrete — resulting in the determination of task clusters and cluster centroids. Parallel to determining clusters, the so-called "search space" is created. The search space is the area around the to-be-assembled product where a robot could be placed theoretically. Thus, a minimum distance limits the search space, and a maximum distance of the executing robot to the product and the direction of execution. Moreover, the search space is discretised to allow for a numerical examination.

By overlapping the area surrounding the clusters' centroids and the defined search space Favoured possible Base Placement (FBP) are determined, limiting the search space, which will be checked for the best results. At each FBP, the IK is solved to generate reachability vectors, indicating the feasibility of reaching specific task poses from a FBP. These vectors are then combined to identify "valid combinations", ensuring the executability of all task poses. These combinations are evaluated based on performance metrics to determine the most optimal formation of robots, culminating in selecting the 'final base placements' for the robots. The respective steps will be detailed below.

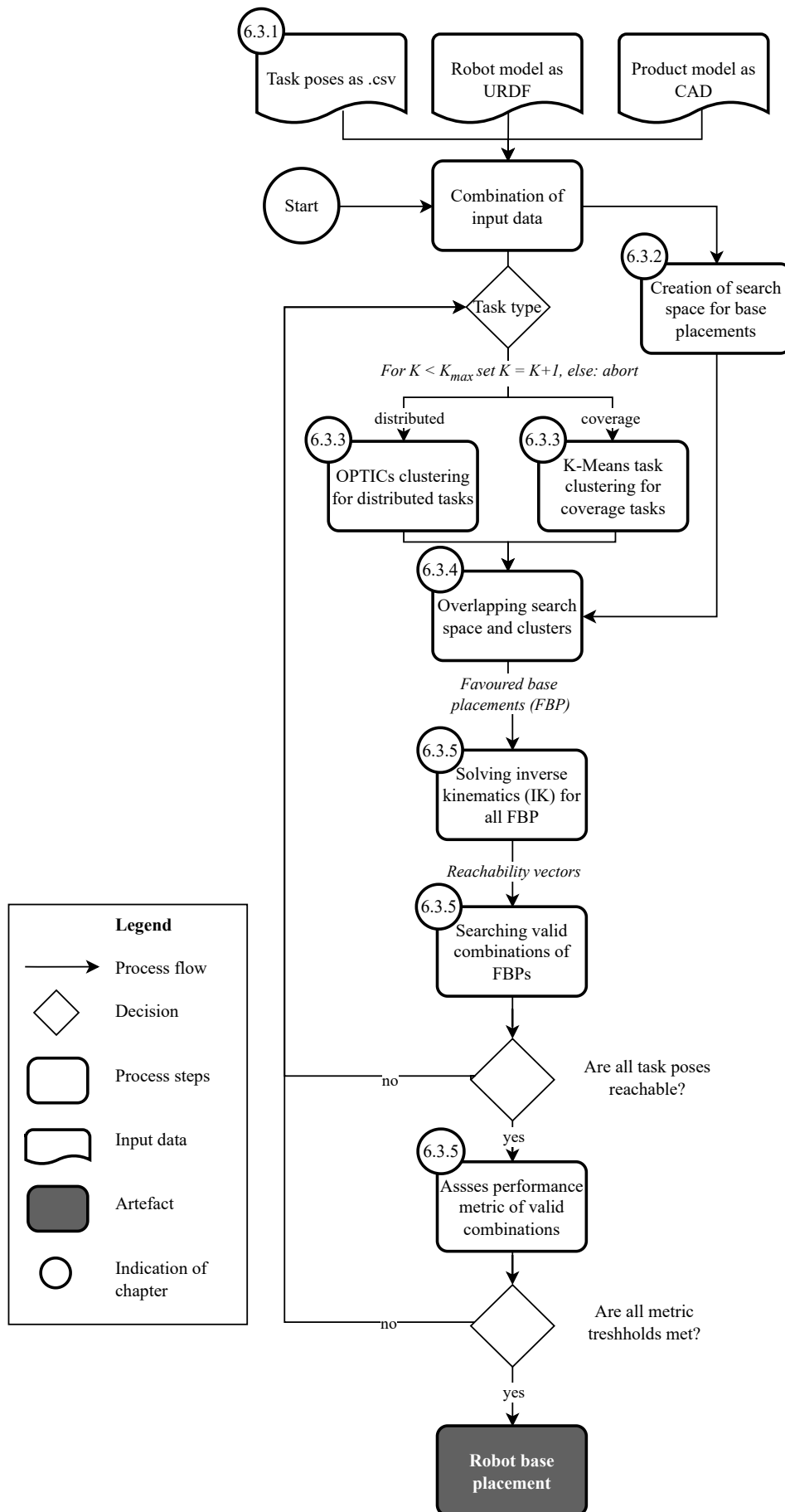


Figure 6.1: Process steps to cluster task poses and determine robot base placements — activity diagram.

6.3.1 Task types

Here, a task is defined as a six-dimensional task pose that specifies the desired position and orientation for the end-effector's TCP during the execution of the assembly task. As introduced in chapter 4 and 5, different task types exist. As for executing different task types, different combinations of resources (an end-effector, robot arm, and mobile base) might be needed. Therefore, these tasks cannot be executed directly after each other without a change of equipment. Thus, here, only a set of tasks as a collection of task poses of the same task type, pertaining to the same industrial operation to be executed on a single product, are considered as input for the autonomous clustering-based task allocation and robot base placement. The product is the to-be-assembled object for which the task poses are provided. When all tasks associated with a product have been completed, the product is finished and can be forwarded to the next station. The task poses are assumed to be given as .csv files during the industrial assembly of products.

The tasks are classified into *discrete* and *coverage* tasks as task types. Discrete tasks can be independently executed, whilst coverage tasks define a number of tasks that must be executed in order. For example, screwing screws would be classified under distributed tasks, while scanning or painting would be classified as coverage tasks.

There are seven attributes associated with each task point n , including the position in x, y, and z coordinates, the orientation as rotation around x, y, and z axes, and a type identifier (*discrete* or *coverage*). The attribute *type* distinguishes between *discrete* and *coverage* tasks and is either given, chosen in dependence on the task's specifications (see section 5.1 and tables 5.1, 5.2). The position values are given in millimetres, while the user can specify the unit of rotations as either degrees or radians. Figure 6.2 provides an example of a CAD model of a turbine and extracted task poses, which shall serve as an example scenario for task clustering.

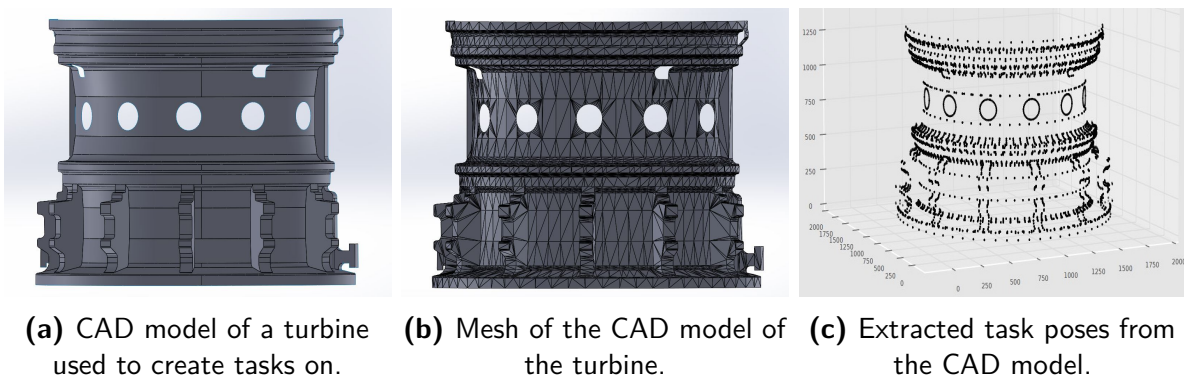


Figure 6.2: CAD model of a turbine and the extracted mesh to create task poses.

6.3.2 Search space

The search for base placements requires selecting a subset of the points on the shop floor around the product poses for the mobile robots while executing a subset of the task poses allocated to this base placement. A search space is created around the product to reduce the amount of investigated points and make the search less computationally intensive.

The search space is limited by removing the areas too close to the product (minimum distance). The minimum distance is set to half the length of the diagonal of the mobile base. Placing a base closer to the product than the minimum distance would be impossible, as the mobile base would necessarily or highly likely collide with the product. Additionally, the search space is limited by excluding areas further than the maximum distance from the product, which is set as the robot's reach. If the robot is placed beyond its maximum reach, it cannot perform the tasks. The minimum and maximum distance used to compute the inner and outer bounds of the search space are represented in red and blue, respectively, in figure 6.3 b).

Additionally, the direction in which the tasks must be executed is considered. Formulated colloquially, the robot has to be placed in front of the product. Thus, the base placements behind the product can be discarded from the search space, as depicted in figure 6.3 c). If the robot is placed behind the product or on the opposite side of the tasks it is allocated to at that moment, the robot would be, in the best case, hindered during execution or, in the worst case, blocked from reaching the task poses by the product itself. Therefore, it is assumed that the robot should be placed in the direction of the surface normal of the product on which the task is to be executed to ensure executability.

If several products are assembled at once, the procedure is done separately for each product. Afterwards, all search spaces are overlapped, and areas within the minimum distance of all products are removed to create a combined search space which does not violate the established boundary conditions, as visualised in figure F.2. Figure 6.3 visualises the steps of creating the search space around the product for a continuous task covering the entire product surface.

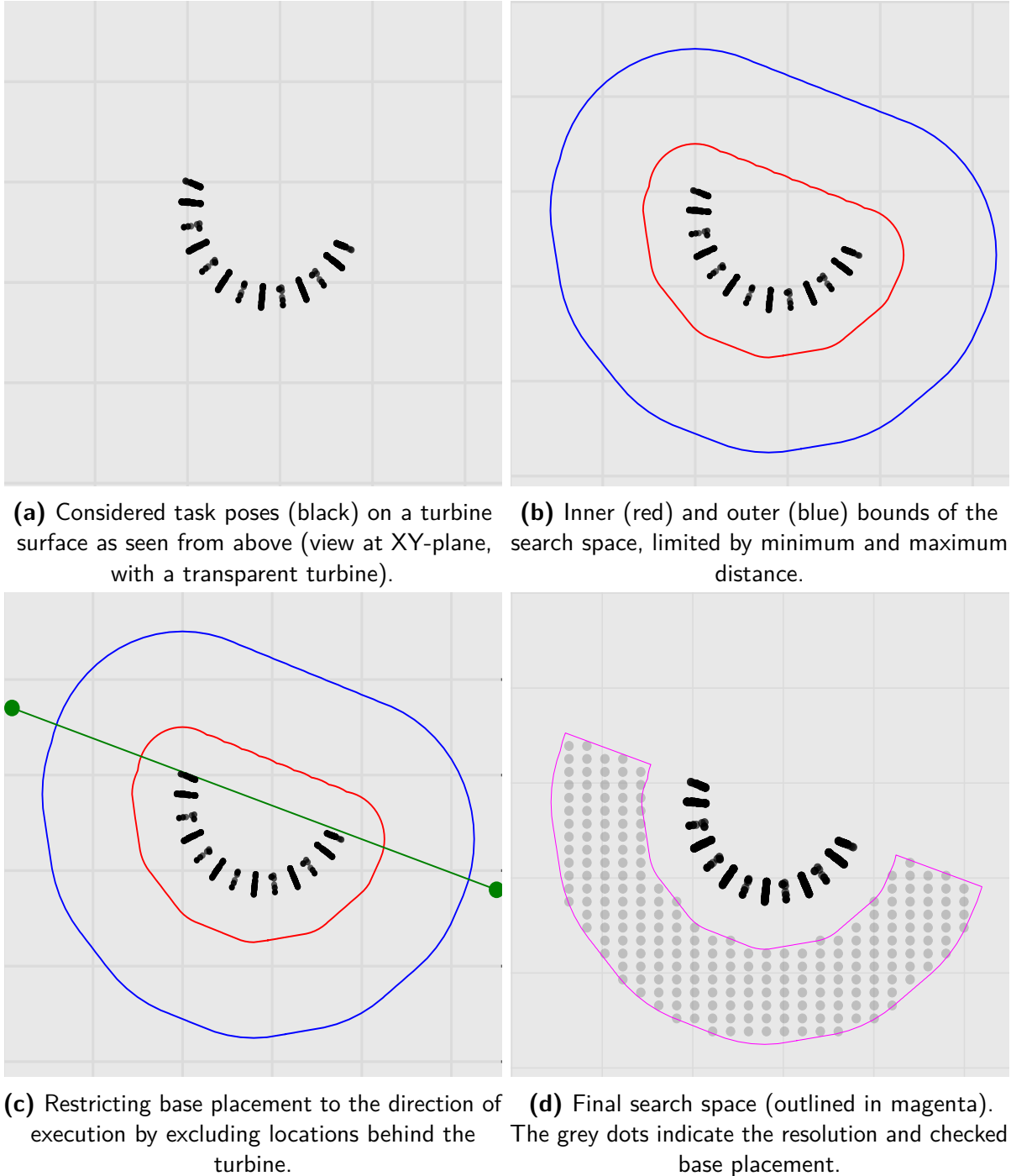


Figure 6.3: Creation of a search space for possible base placements around the product.

6.3.3 Task clustering

When the entire set of tasks can not be performed from a single robot's base placement, it must be divided into smaller sub-tasks. Clustering can automatically create such sub-tasks by clustering the task poses based on their distance.

Intermediate design cycles applied clustering as the only means of deriving base placements, but clustering does not consider obstacle avoidance and possible collisions. This can be seen from the centroid placement indicated as black crosses in figure 6.5. The centroids are placed inside of the turbine on which the task poses are placed. Thus, the clustering method cannot

be applied as stand-alone and has to be combined with a heuristic approach of limiting the area of allowable base placements. Therefore, the previously established method of defining an allowable area for base placements, the search space, was derived. Here, task clustering is applied to reduce the number of base placements resulting from the discretisation of the search space.

As defined in section 6.1 *all task poses have to be allocated to a cluster*, since the execution of all assembly tasks is necessary for a successful product assembly. Therefore, only hard/crisp clustering algorithms can be applied, as they ensure that every task is assigned to one cluster. During the first design cycle, K-Means as the clustering method was investigated. Since K-Means is suitable for densely spaced tasks but not for far-distributed single tasks, coverage tasks and discrete tasks are differentiated, and the second clustering method OPTICS is introduced in the second major design cycle. The K-Means clustering algorithm finds clusters and centroids of sets of coverage tasks, whilst OPTICS is applied for discrete tasks.

The following details how K-Means is incorporated into the autonomous clustering-based task allocation and robot base placement and which parameters are chosen. The aim is to adapt K-Means in such a way that the size of the clusters is similar to that of a robot workspace. K-Means requires a set number of clusters K , and the data points to cluster as an input. As the number of clusters is unknown, the K-Means clustering step is iterated and the number of clusters is increased in each iteration until all task poses are allocated into clusters, reachable, and the metric thresholds for the assignment are met (see figure 6.1). The user has to set an upper limit of allowable clusters K_{max} , ensuring that the autonomous clustering-based task allocation and robot base placement do not run infinitely if the task poses do not match the requirements.

The clusters are optimised according to the silhouette analysis during each K-Means iteration cycle. The silhouette coefficient provides a metric of similarity of a data point to the other points within a cluster compared to those data points not in the same cluster. The equation 6.1 calculates the silhouette coefficient $C(e)$ for each pose e .

$$C(e) = \frac{b(e) - a(e)}{\max[a(e), b(e)]} \quad (6.1)$$

where $a(e)$ is the average distance between pose e and all other poses in the same cluster, and $b(e)$ is the average distance from pose e to all poses not in the same cluster. The internal distance of data points within a cluster is minimised, and the average distance within each cluster is balanced regarding the average distances of the other clusters. The silhouette score for a cluster is the average of all silhouette coefficients within the cluster. It ranges from -1 to 1, with 1 indicating the highest cohesion.

Discrete task poses are clustered by the clustering method OPTICS. The autonomous clustering-based task allocation and robot base placement method then chooses OPTICS as the clustering method for the discrete tasks and calculates clusters based thereon. The task poses are given as inputs.

As detailed in section 6.2, the integer *minSamples* has to be given and corresponds to the allowable minimum size of a cluster regarding its assigned data points. If a singular discrete task can not be included in any clusters because the distance to the nearest neighbour is too great and thus exceeds the reach of a single robot, it will be assigned to its cluster. Therefore, *minSamples* has to be set to one.

Compared to K-Means, OPTICS does not assign centroids to found clusters. Following the clustering, the same approach shall be used for discrete and coverage tasks. Thus, a centroid is created for clusters derived by OPTICS by taking the mean value of the positions of all assembly task poses of one cluster as a centroid.

6.3.4 Favoured base placements

The upcoming section outlines estimating favoured base placements from the previously derived search space surrounding the product and the clusters of task poses. The search space is discretised equidistantly to reduce the number of possible base placements to test. The distance between each discrete point in the search space is hereafter named resolution. Figure 6.4 visualises augmenting the resolution. Black dots represent the task poses. The product is enclosed by magenta lines delineating the search space's boundaries, and the discrete possible base placements are depicted in grey. While a higher resolution is generally expected to return base placements that are more likely to fulfil the desired outcome of reachability of all allocated task poses in the cluster, it also increases computational complexity.

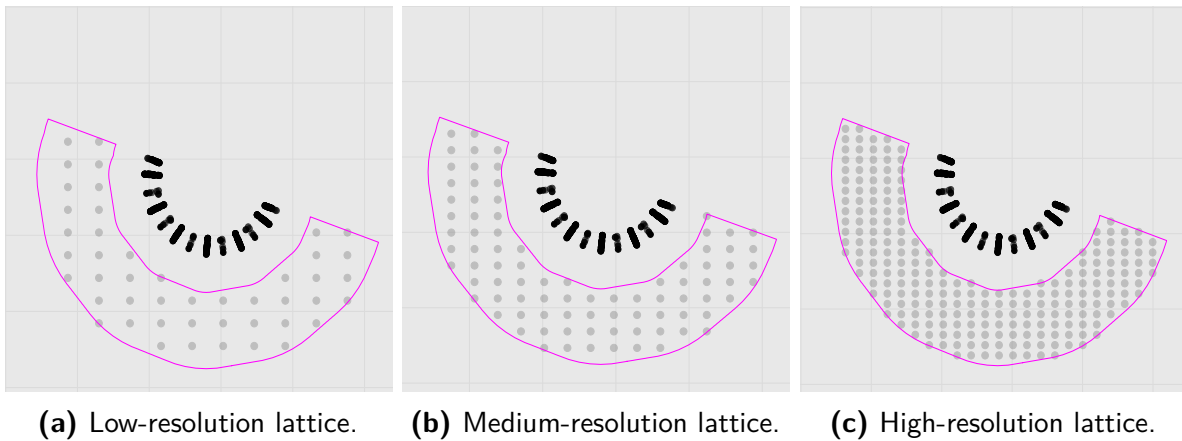


Figure 6.4: Visualisation of the same search space discretised with increasing resolution. The black dots indicate the task poses on a turbine as seen from above, the magenta line outlines the search space, and the grey dots indicate the number of discrete base placements as resolution.

The resulting centroids of either clustering method (see section 6.3.3) for both types of assembly tasks are taken as starting points to identify FBPs by overlapping the area around one centroid with the discrete points of the search space (see section 6.3.2). A circle with a radius equivalent to the maximum reach of the robot is generated around the centroid of each cluster, thereby ensuring the reachability of the task poses. Poses within both areas (discretised search space and circled centroids) are designated as FBPs for a specific cluster. The search for the final

base placements for each cluster will be confined to these points. Section 6.3.5 explains the methodology employed in this search.

The geometric realisation is visualised in figure 6.5. The task points on the product turbine are clustered into three clusters (red, blue and yellow dots). The three black crosses mark the centroid of each cluster. For each cluster, a circle with the radius of the maximum robot reach is projected over the area, with each cluster's centroid as the circle's centre. The overlap of the circle with the discretised search space results in the discrete possible base placements (marked as green dots). The boundary of the search space is visualised as a magenta line, and the discrete points of the search space, which are not taken as possible base placements, are visualised as grey dots.

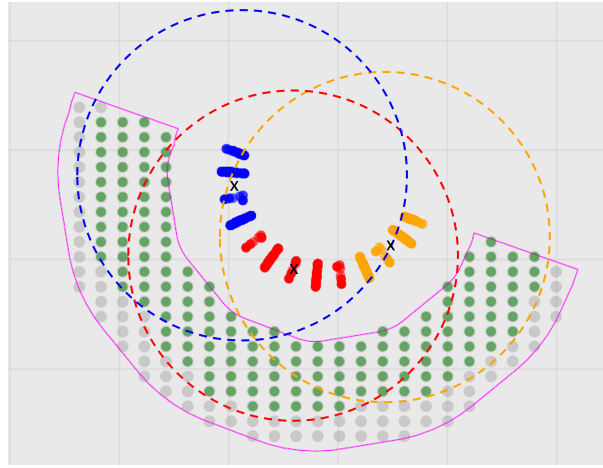


Figure 6.5: Visualisation of the extraction of favoured robot base placements. Three task pose clusters are colour-indicated (blue, red, and yellow dots); the black x indicates the respective cluster's centre. For each centroid, the coloured circle indicates the maximum robot reach from this centroid. The previous search space (grey and green dots) is reduced to the green dots by overlapping the circles with the previous search space (magenta line).

6.3.5 Selection of base placements

Based on the calculation of favourable base placements, the next step is to find a combination of FBPs that ensures the reachability and executability of all task poses.

A so-called reachability vector is created for each favoured base placement. Reachability vectors are 1-dimensional arrays consisting of ones and zeros. They designate whether a task pose can be reached from a possible base placement (1) or not (0). They are of length n , corresponding to the total number of task poses n . The IK of the robots placed at the favoured base placements to reach all task poses in their assigned cluster is solved to build the reachability vector. Moreover, the region of feasible robot flange poses is calculated based on the end-effector and the task type (as explained in section 5.2.1). One example for the calculation of the regions of feasible robot flange poses is visualised in figure 6.6.

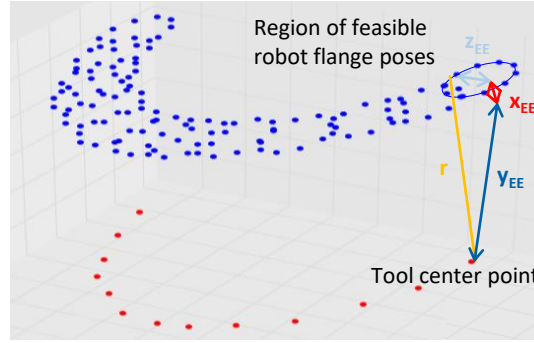


Figure 6.6: Visualisation of the regions of feasible robot flange poses for a screwing task. The red points indicate the task poses, the blue circle indicates the region of feasible robot flange poses, and the blue dots represent discretised poses of the region of feasible robot flange poses for each task pose. Based on the screwing task illustrated in figure 5.6.

Additionally, the chosen performance metrics (e.g. manipulability measure, dexterity index or condition number as explained in section 2.2.2) are calculated for each discretised pose of the region of feasible robot flange poses. As only solutions with a high manipulability index are chosen, the executability and existence of trajectories to execute a sequence of task poses are likely, accounting for HR06.

The reachability vectors are searched for a combination ensuring reachability of all task poses from at least one base placement ("full reachability"). Vector combinations with only non-zero elements ensure full reachability. Thus, implicitly, HR02 and HR03 are met. The number N_{rv} of combinations that need to be searched is given by equation 6.2, where K_{max} is the maximum number of base placements and k is the current number of base placements. Combinations that return full reachability are stored as valid combinations, and the rest are discarded. Thus HR04 is fulfilled, by ensuring that each task pose is reachable from at least one base placement.

$$N_{rv} = \sum_{k=1}^{K_{max}} \frac{K_{max}!}{k!(K_{max} - k)!} \quad (6.2)$$

Once the valid combinations of possible base placements have been identified, spanning the global solution space, they are evaluated based on several criteria to determine an optimised solution. These criteria include minimising the number of base placements (SR01), maximising the distance between base placements (SR03), and maximising the performance metrics (HR06, SR05).

It is assumed that a minimised number of base placements either correlates to a minimised amount of robots working on the same product at the same time, thus reducing the amount of needed hardware, thereby minimising the cost of operation, or the minimised number of base placements correlates to a minimised amount of base placements changes for a single robot, thus reducing the operation time. Therefore, a minimised number of base placements is considered beneficial. Thus, valid combinations with the least amount of base placements are rewarded (HR05, SR01).

To minimise the chances of collisions between concurrently operating robots (SR03) and to avoid frequent repositioning of robot bases, it is advisable to position the robots as far away from each other as feasible. To accomplish this, the average Euclidean distance between the base placements of each valid combination is calculated and recorded in a score vector and higher distances are rewarded.

Here, the application of the manipulability measure as a performance metric to evaluate executability shall be detailed, accounting for SR04, as solutions with a high manipulability value are not close to singularities. Generally, other chosen performance metrics (as detailed in chapter 5) can be included similarly. To avoid singular robot configurations, which occur when the manipulability measure equals zero (as explained in section 2.2.2), a high manipulability measure is rewarded. Each valid combination is scored according to the objectives (minimised number of base placements, maximised distance between base placements, and maximised performance metrics) in score vectors s_n . s_n is given as equation 6.3.

$$\begin{pmatrix} s_1 & s_2 & \cdots & s_n \end{pmatrix} \quad (6.3)$$

The score vectors of all valid combinations are combined into the score matrix S , as generalised in equation 6.4.

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{pmatrix} \quad (6.4)$$

The scores can be rewarded or penalised with the weight vector W , shown below in equation 6.5. Therein, the weights represent the priorities of a minimised number of base placements, the distance between base placements, and the applied performance metrics.

$$W = \begin{pmatrix} w_1 & w_2 & \cdots & w_n \end{pmatrix} \quad (6.5)$$

Multiplying the score matrix S and the weight vector W results in the weighted scores vector shown in equation 6.6.

$$SW^T = \begin{pmatrix} \sum_1^n s_{1,n} w_n \\ \sum_2^n s_{2,n} w_n \\ \vdots \\ \sum_m^n s_{m,n} w_n \end{pmatrix} \quad (6.6)$$

The result with the highest score among all valid combinations is selected as the final base placement. If multiple robots can reach a task pose, it is allocated to the robot whose base placement is closest to the task pose, whereby the distance is calculated as the Euclidian distance. Thereby, HR01, HR04 and SR02 are accounted for.

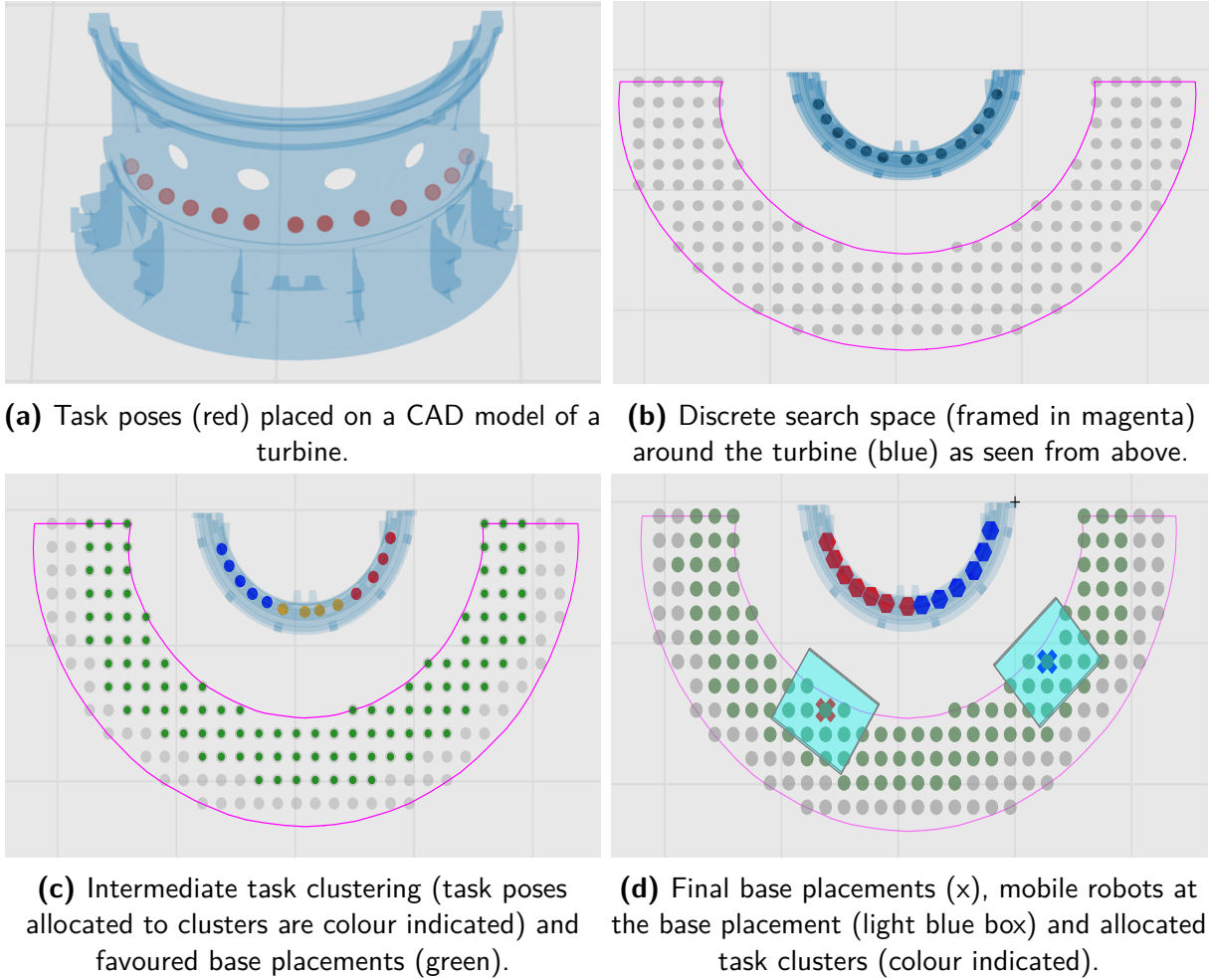


Figure 6.7: Process of deriving robot placements from task poses placed on a turbine's CAD model.

Figure 6.7 visualises the process of discretising the search space around the product and the successive placement of mobile robots around it. At first, the task poses (red in figure 6.7 a)) are placed on a CAD model of a turbine as a base for the practical implementation and testing of the proposed task clustering method. Secondly, the search space is derived (as depicted in figure 6.7 b)) by the maximum reach of the robot as the maximum distance and the size of the mobile base as the minimal distance from the turbine. The magenta line indicates the search space. The grey circles indicate discrete positions, which form the number of positions to be checked for robot base placements. Thirdly, the favoured base placements are identified, and task points are clustered, as shown in figure 6.7 c). The blue, yellow, and red marked task poses indicate three different clusters as an intermediate result of the iteration that minimises the number of clusters. The green circles indicate favoured base placements, whilst the grey circles indicate discrete base placements that have not been chosen in the search space. In the fourth step, the final two robot base placements are chosen (figure 6.7 d)). The blue boxes indicate the top view on the placed mobile bases, with the robots on top. The red task poses on the turbine are allocated to the mobile robot standing on the red marked base placement, and analogous, the blue task poses are allocated to the robot on the blue marking.

6.4 Verification and validation of the base placement and task clustering

To validate and verify the presented research artifact of autonomous clustering-based task allocation and robot base placement, its usefulness is evaluated. The artifact is designed to address the unsolved problem of assigning places to mobile robots in changeable assembly systems. Consequently, the artifact is applied in exemplary assembly scenarios, and the outcomes are examined by testing the executability of all tasks within a single cluster through the assigned robot. The subsequent section summarises the results of this evaluation and validation. In addition to the known application example of turbine inspection, the application scenario of I-Beam inspection and welding points on the I-beams are introduced. Thereby the inspection tasks serve as examples for continuous tasks, whilst the welding serves as example for discrete task poses.

Initially, the systematic verification and validation methodology is outlined. Next, the test scenarios for validating and verifying both the solution method and its results are explained. The artifact of autonomous clustering-based task allocation and robot base placement undergoes testing by application, and the experiment outcomes are presented and discussed. The underlying verification and validation methodology is explained in section 2.5.

Data validity

The assumption that task poses are given as .csv files was confirmed as achievable in several discussions with experts from different industries (automotive, battery production, and timber house production)³⁴. But, for the purpose of testing the artifact of autonomous clustering-based task allocation and robot base placement, no industrial data sets in the required form of task poses as .csv files could be acquired from industrial partners. Since no task poses from industrial use cases were prevalent, artificial use cases were created. The creation of data sets aimed to be close to realistic industrial applications and to cover a broad field of application scenarios to ensure the applicability of the proposed clustering method to different assembly tasks. Four methods were applied to create exemplary tasks:

1. Manual creation of task poses on a CAD file and extraction of these.
 2. Partially automated creation of distributed task poses on a random number of products.
 3. Converting a CAD file into a mesh file, isolating the vertex points of the mesh triangles, and then choosing a specific subset of those points.
 4. Extracting poses on a test object with a metrology system.
1. As an example of the manual creation of task poses, figure 6.8 a) visualises the constructed part of an I-beam as well as the manually placed task poses to create a *continuous task*. Reference poses are positioned by placing them on lines drawn on the I-beam's upper surface. Lastly, the reference poses are extracted from SolidWorks and saved as a .csv file. The created

³⁴Even though it is not current practice to provide task poses as .csv, but, e.g., as a robot program running on its controller or in proprietary file formats, the process of converting into .csv is seen as achievable.

CAD file of turbine and I-beam are used as an exemplary product, but the procedure could be extended to any CAD model.

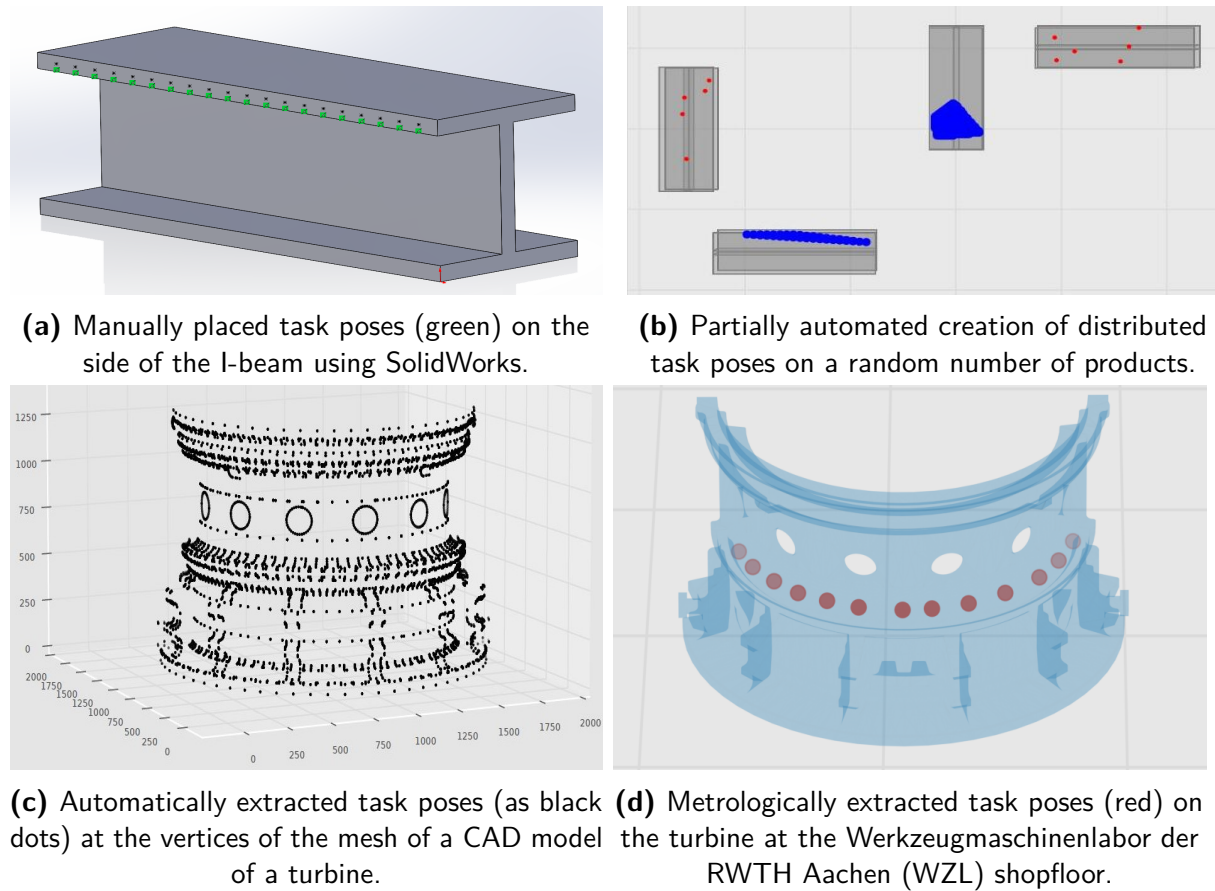


Figure 6.8: Results of different methods to create exemplary task poses on products.

2. The partially automated creation of distributed task poses on a random number of products relies on manually specifying the number and placement of I-beams, as well as the number and type of tasks (continuous or discrete) on each beam. Then, the Python standard library "random" [VAN 09] pseudo-randomly selects coordinates within the vertices of the surface of each I-beam and places the tasks on those coordinates, as visualised in figure 6.8 b). Further examples can be found in the appendix in figure F.1.

3. Figure 6.2 visualises the process of converting the CAD file into a mesh to extract the poses of the vertices. A .stl file format is used to save the mesh, as it can be read by any Python library that operates on this file format. The numpy-stl library extracted the poses from the mesh. The process reads the vertex coordinates of the triangulated surface, as described in [HATT22]. Depending on the resolution of the .stl file, many task poses will be created with this automated task pose generation. Based on this method of generating task poses, the behaviour of the proposed task clustering can be tested concerning behaviour for extensive input. Figure 6.8 c) illustrates the end result with all extracted task poses.

4. Aiming at real-life validation of the proposed task clustering, the fourth method of creating task poses by touching poses on the physical object of a turbine with a metrology system, then transferring those poses into goal poses for the robot end-effector, is utilised. Here, the

so-called iGPS system was utilised. For additional information on the metrology system see [23a]. The metrology system iGPS outputs a coordinate in a world coordinate system. This world coordinate system serves as a reference for the mobile robot. The resulting test setup is shown in figures 7.1 and 7.2 f), whilst the resulting take poses are mapped to the CAD model in figure 6.8 d).

Therefore, the utilised data is valid according to SARGENT (2010).

Conceptual model validation

The validation of the conceptual model is crucial to confirm the suitability of the task clustering algorithm for base placement. The conceptual model is validated by continuous comparison with the requirements set out at the beginning in section 6.1. This continuous verification and validation during the iterative design cycles is based on the techniques of animation and face validity. To ensure a broadened view during **face validation**, feedback from the research group "Future Assembly" of the research project IoP and feedback from reviewers of publications were included. Feedback and discussion sessions were conducted during the publication process of [KLUG23a]. This comprehensive approach to validation ensures the integrity and reliability of the task clustering algorithm for base placement.

As can be seen from the pictures taken and shown to visualise the undertaken verification and validation tests, the validation technique of **animation** was applied to ensure satisfactory system behaviour. During development as well as for testing purposes, the task poses, the objects on which the task poses are placed, the search space resolution and the resulting task clusters and allocated base placement for each cluster are visualised. This verification and validation method enables quick and intuitive assurance of proper system behaviour.

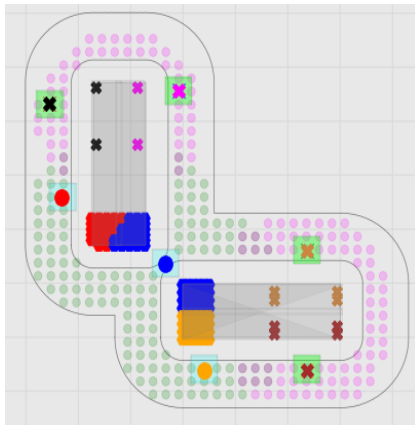
Computerised model verification

The computerised model verification ensures the correct implementation and programming of the designed model. Verification is confined to internal considerations, excluding the real-world aspect, while validation incorporates external real-world factors.

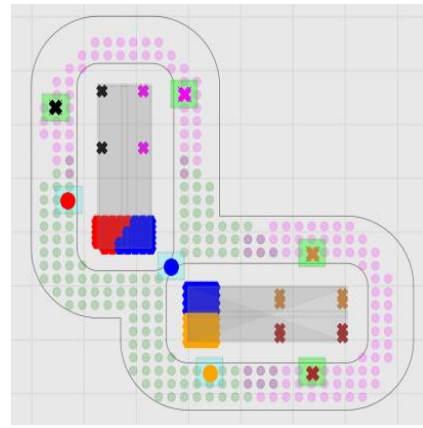
Static and dynamic testing was applied to the implemented executable program. As static testing, structured walkthroughs of the implemented code were conducted to verify the correct implementation of the underlying modelling. The dynamic testing techniques include animation and internal replication. Animation entails visually representing the temporal sequences. Its utility lies in analysing the plausibility of a process or its outcomes. Animation is effective for identifying and correcting invalid situations. As the artifact of autonomous clustering-based task allocation and robot base placement operates as a decision model rather than a simulation model, the animation scrutinises the optimisation process and showcases the planning results. This aims to detect any inconsistencies or contradictions with the intended decision behaviour. Inherent to the artifact is the frequent visualisation of the intermediate clustering results, each step of defining the search space and the final base placements.

To ensure the **internal validity** of the proposed algorithm for task clustering and base placement, the consistency of the solution method's results is tested by replicating the same scenario multiple times. For internal validity to hold, the algorithm should return similar results for similar inputs.

The same test case (as depicted in figure 6.10) is taken as input for all test runs whilst changing the resolution of the search space. Each created test set is solved three times, and the computational time is indicated. The tests were run on a Dell Inspiron 15-7577 with Intel(R) Core(TM) i7-7700HQ CPU and 16 GB of memory running on a Windows 10 operating system. Across three test runs with identical resolution and task poses, the time difference varied from 3.65% to 17.2%, with an average of 10.71%, as depicted in table 6.1. This discrepancy can be attributed to the time required for the numerical method to discover an inverse kinematic solution, as well as the background processes that the operating system performs automatically without user input.



a) Resolution: 961, Completion time: 908.1 seconds.



b) Resolution: 961, Completion time: 936 seconds.

Figure 6.9: Identical base placements (light blue and green rectangles) and time variance for executing similar test scenarios multiple times. The grey rectangles indicate the product, the green circles indicate possible base placements for coverage tasks, and the pink circles indicate possible base placements for discrete tasks. The task poses are allocated to clusters as indicated by different coloured dots.

Resolution [mm]	Time 1 [s]	Time 2 [s]	Time 3 [s]	Maximal time deviation
280	448.8	448.6	432.2	3.65%
270	694.3	707.5	670.2	5.27%
260	616.7	608.9	714.1	14.8%
250	751.5	766.1	860.4	12.67%
240	1180	977.1	1002	17.2%

Table 6.1: Resulting time variance from executing similar test scenarios with the artifact of autonomous clustering-based task allocation and robot base placement multiple times.

As seen in figure 6.9, repeating the test scenario multiple times results in identical base placements. The green circles indicate FBP for the task poses belonging to continuous task sets, whilst the magenta circles belong to the task set containing discrete task poses. The

crosses mark the final solution for the base placements for the discrete task poses, whilst the framed and highlighted circles mark the final solution for the base placements of the continuous tasks. The respective colours of the base placements indicate the assignment to the task poses, which are marked in a similar colour.

Operational validation

Operational validation ensures the correct model is built to answer the research question and satisfy the stated requirements. It aims to test the accuracy of the model's output behaviour, so the testing is limited to the aspired domain of the application. Here, the intended use of the developed algorithm is the base placement of mobile robots.

As RABE ET AL. (2008) stated, the process of validation includes the comparison of empirically collected data from the actual system with data provided by the abstract model [RABE08, p.15].

A **sensitivity analysis** was conducted to assess the parameter variability by altering the input parameters, such as search space resolution and the number of task poses, to measure the impacts on the behaviour of the solution. In the initial test, the resolution of the search space was increased while maintaining a constant number of task poses (e.g. as visualised in figure 6.4). The subsequent test involved raising the number of task poses while keeping the search space resolution constant.

The algorithm was executed on a scenario involving two I-beams, with both continuous and discrete tasks, while gradually increasing the resolution of the search space. The tasks are depicted in figure 6.10, with the coverage task represented by blue spheres and the discrete task represented by red crosses. The coverage task includes 95 poses, while the discrete task comprises 24 poses.

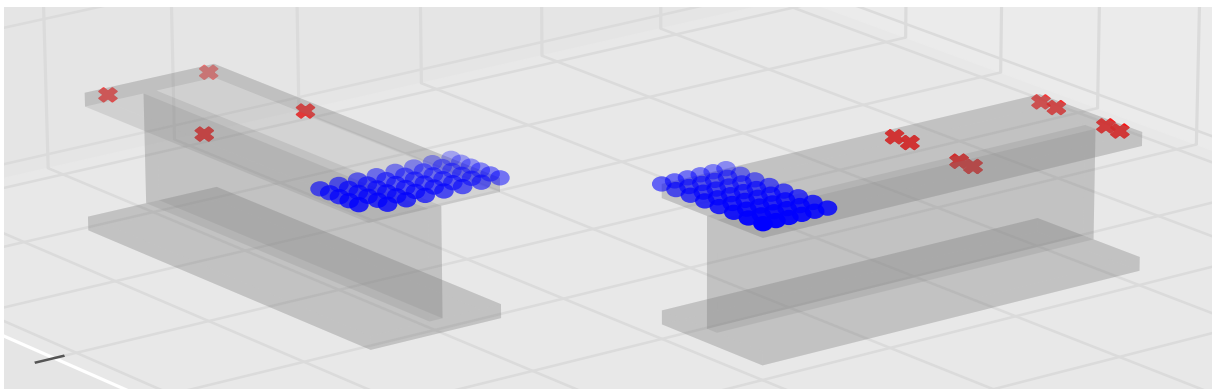


Figure 6.10: Assembly task poses on products: Test scenario with continuous task poses (in blue) and discrete task poses (in red) distributed on two I-beams.

Surrounding the two I-beams, the search space, the area for possible base placements, is created. To avoid collisions, the maximum reach of the robot is limited to the maximum distance from the I-beams, and the size of the mobile base is limited to the minimal distance from the I-beams.

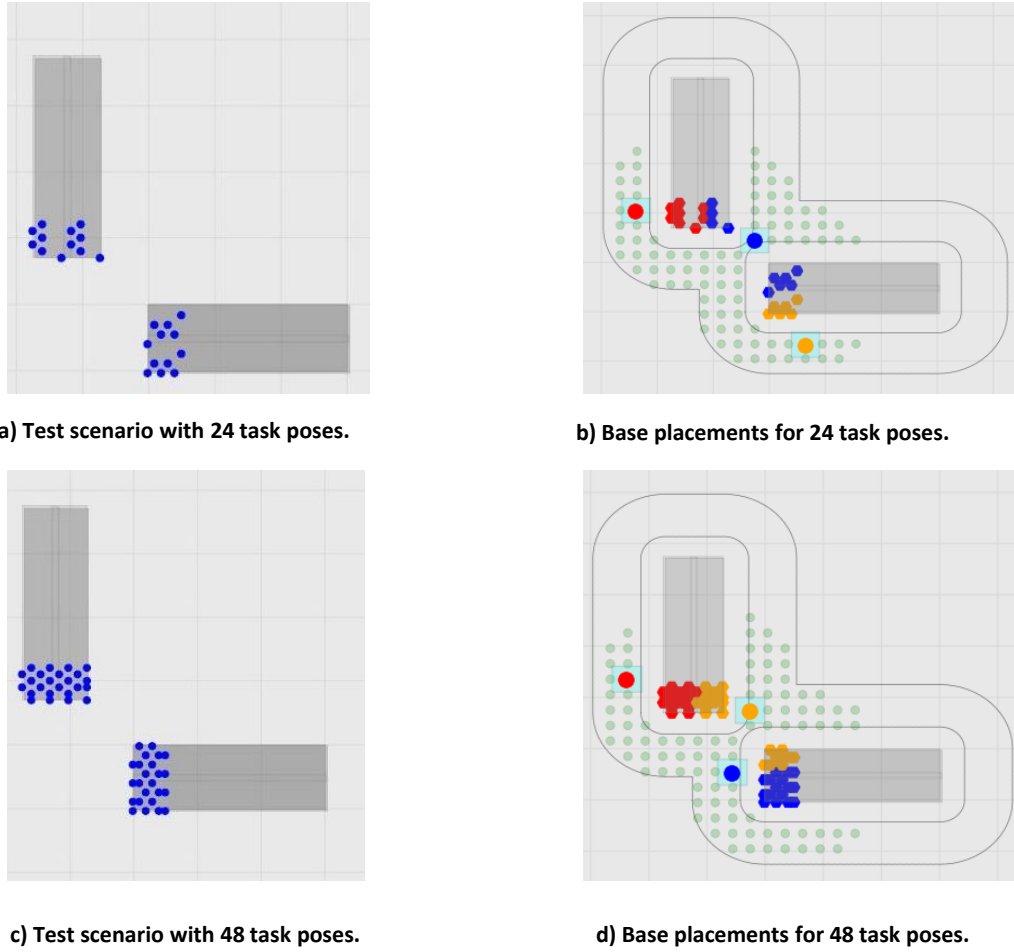


Figure 6.11: Top view of the test scenarios with identical search space resolution and increasing task poses. Figures a) and c) show the input: the product in grey with task poses represented by blue dots. Figures b) and c) visualise the respective found base placement solution. Potential base placements are highlighted by green dots within the search space delineated by grey lines. The found base placements and allocated task pose clusters are differentiated by various colours.

Scenarios, where the resolution of the search space is too low, may lead to the absence of feasible solutions, resulting in task poses that cannot be executed from any base placement. Exemplary scenarios are visualised in figure F.4 in the appendix. Increasing the resolution generally results in fewer base placements, which is seen as beneficial since it requires less movement between the base placements for the robot, as depicted in the appendix in figure F.2. But simultaneously, a higher resolution results in a higher computation time, as visualised in figure 6.12 c). The relation of search space resolution (as the number of base placements), the achievable score (as a quality measure of the final solution) and the needed computation time is visualised in figure 6.12 d). As can be seen, the score of the finally found base placements reaches a plateau for a certain amount of base placements. Thus, infinitely increasing the resolution and, thus, the amount of checked base placements is not advisable. This plateau has to be found as the highest possible score is favourable.

Since a high number of task poses may occur in an industrial scenario, e.g. when a large object should be inspected with a high resolution of task poses, the algorithm's behaviour is

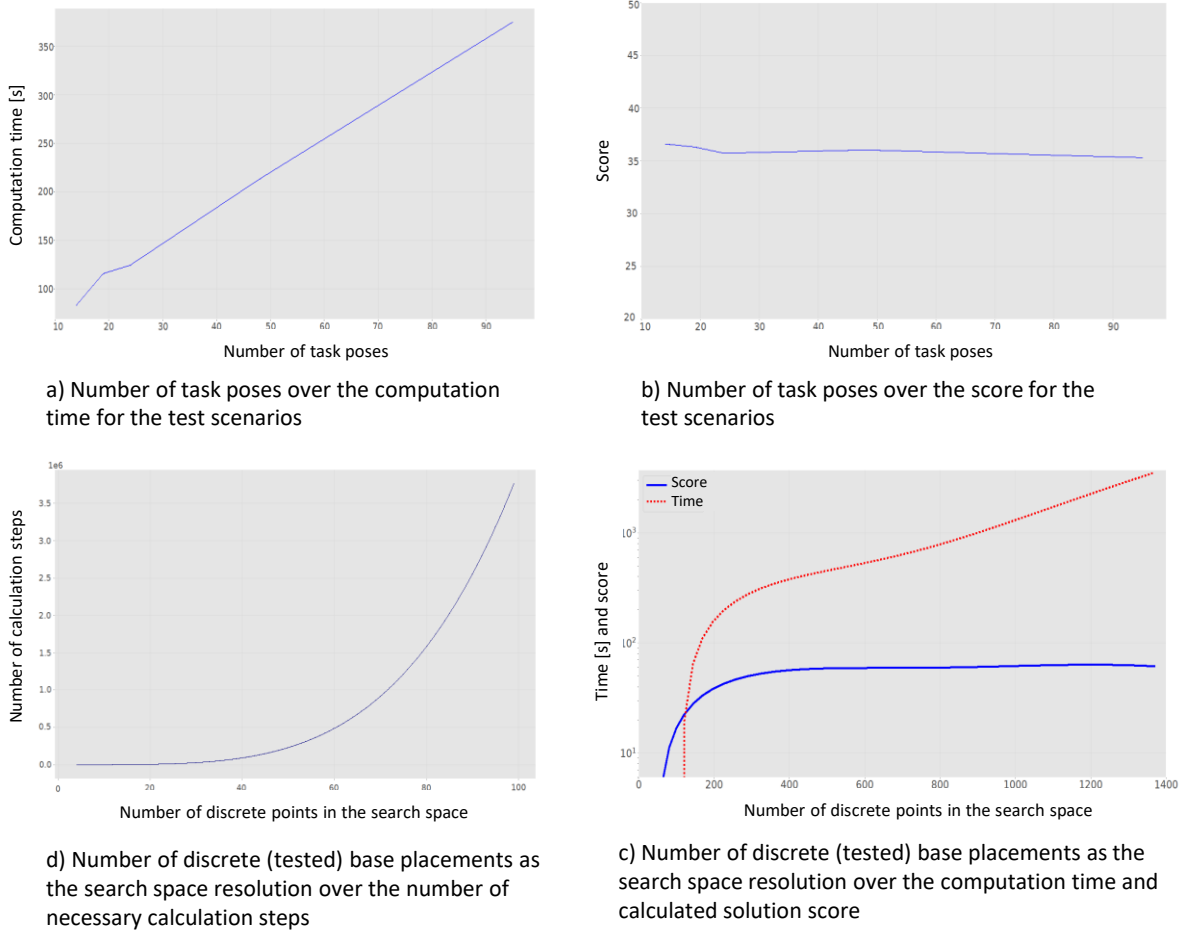


Figure 6.12: Results of the autonomous clustering-based task allocation and robot base placement when increasing the number of tested task poses as part of the operational validation.

investigated concerning handling increasing numbers of task poses. Therefore, the search space resolution is set at a fixed number, and tests are run with increasing task poses, placed on a similar area on the product. Exemplary test scenarios can be seen in figure 6.11, as well as the corresponding results of the base placement. Blue dots on the left side mark task poses on the product's surface. On the right side, the possible base placements are marked in light green, whilst chosen base placements and allocated task poses are marked in red, blue and yellow (further examples are in the appendix in figure F.3). A solution (a number of base placements for the robots) that covers the entire task set was found for every test scenario. Figure 6.12 b) illustrates that the quality of the solution remains mainly unaffected by an increase in task poses, as the maximum deviation between the scores given to each solution is only 3%.

6.5 Interim conclusion: Clustering-based task allocation and base placement

This chapter presents a method for autonomous clustering-based task allocation and robot base placement, addressing the third underlying research question URQ3: ***How can assembly tasks***

be clustered to plan formations with reduced base placements of mobile assembly robots?

The question is answered by the presented autonomous clustering-based task allocation and robot base placement method. The multistaged heuristic identifies a minimised number of base placements — thereby enabling the selection of an **optimised assembly station formation**. In the first stage, the search space — the confined area of allowable base placements around the product, ensuring reachability — is created, parallel to grouping the task poses into clusters, executable from a single base placement. By overlapping both solutions, favoured base placements are found. The favoured base placements are searched for combinations, minimising the total number of base placements whilst guaranteeing the reachability of all task poses. Finally, the combinations found are evaluated for executability by applying performance metrics. The derived heuristic identifies and **evaluates formations**, whereby a formation is defined as the allocation of task poses to mobile robots placed at specific base placements. The developed method is tested in both simulation and real shopfloor environments. The optimisation of mobile manipulator base placements proves to be an effective means of enhancing both the efficiency and quality of task execution. The chapter's contribution is summarised in figure 6.13.

The following table 6.2 matches the initially defined hard requirements (see section 6.1) with the applied solution. The soft requirements are met by selecting a set of valid solutions that fulfil the hard requirements. The soft requirements are quantified and included as scoring criteria and weights as described in section 6.3.5. The solution with the highest score is chosen. Thus, the soft requirements are met.

Chapter 6 contributes to the research artifact of autonomous clustering-based task allocation and robot base placement. The artifact is added to the knowledge base as prescriptive knowledge in accordance to [IIVA07, p.8].

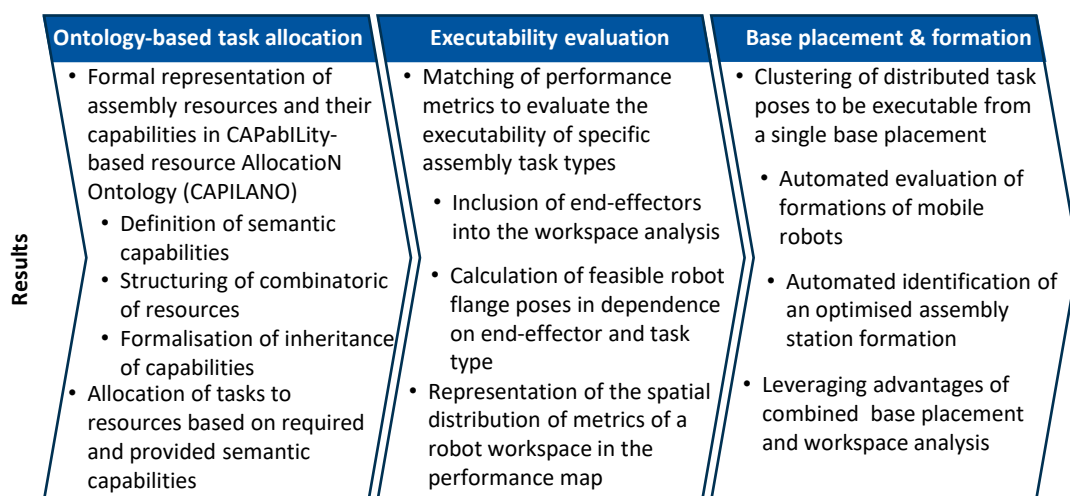


Figure 6.13: Contribution of chapter 6 to the research progress.

Table 6.2: Matching the requirements (derived in section 6.1) with the applied solution (detailed in section 6.3.5) of clusterin-based task allocation and base placement.

Reference	Requirement	Applied solution	Met?
HR01	All task poses have to be allocated to a cluster.	Application of the clustering algorithm (either K-Means or OPTICS, in dependence of the task type (<i>continuous</i> or <i>discrete</i>)).	✓
HR02	For each cluster, a valid base placement has to be found.	Reachability is checked by solving IK, and only combinations with "full reachability" are accepted as valid combinations.	✓
HR03	A base placement is valid if all allocated task poses of the cluster are reachable by the robot placed at the base placement.	Reachability is checked by solving IK, and only combinations with "full reachability" are accepted as valid combinations.	✓
HR04	It follows: all task poses must be reachable from at least one base placement.	Reachability is checked by solving IK, and only combinations with "full reachability" are accepted as valid combinations.	✓
HR05	Collisions with physical objects have to be avoided.	Minimum distance from the product in dependence on the size of mobile base is applied. Minimised amount of base placements ensures a maximised distance of robots from each other. Collisions during dynamic trajectory execution are not considered.	(✓)
HR06	Executability has to be ensured.	Solutions with desirable executability metric values are preferred. High manipulability of valid combinations is rewarded. Trajectories are not considered.	(✓)
SR01	The number of clusters/base placements should be minimised.	valid combinations with the least amount of base placements are rewarded.	✓
SR02	Each task pose should only be allocated to one cluster.	If multiple robots can reach a task pose, it is allocated to the robot whose base placement is closest to the task pose.	✓
SR03	Possibilities for collisions should be reduced.	The average Euclidean distance between the base placements of each valid combination is calculated and recorded in a score vector and is rewarded.	✓
SR04	Avoidance of singularities.	High manipulability of valid combinations is rewarded.	✓
SR05	Consideration of feasible trajectories between task poses.	By choosing solutions with a high manipulability metric, feasible trajectories between task poses are likely, but this is not verified.	(✓)

Critical reflection

The step of validating the reachability and executability of all poses from a set of base placements can be improved upon. The current method depends on checking all possible combinations of base placements and determining the metrics chosen (choice detailed in section 5.1) of all poses in the workspace of a robot placed at a base placement (as described in section 6.3.5). This step can be improved by utilising the calculated performance maps as described in section 5, to reduce the computational time by overlapping the area of the robot's workspace that ensures reachability of all poses inside of this area.

For the creation of the search space, the direction from which the tasks have to be executed in relation to the product CAD and the dependent exclusion of possible base placements behind the product are currently handled manually. This could be automated further to enhance the level of automation of the presented method. Moreover, an upper limit of clusters is currently set manually. An upper limit could be implemented into the autonomous clustering-based task allocation and robot base placement, e.g., by estimating the maximum number of robots fitting into the station simultaneously, calculated by taking the station's area and dividing it by the footprint of the mobile base.

Alternatively to the current implementation as a multistaged heuristic based on clustering, the autonomous clustering-based task allocation and robot base placement problem could be formulated as an optimisation problem. Different solving strategies with established optimisation problem solvers could be applied and compared based on criteria like computation time for solving or optimality of the found solution.

The current implementation does not address potential challenges that may arise during the practical integration of the method into operational industrial applications. The assumption that a high manipulability measure ensures the dynamic execution of all allocated task poses needs verification, particularly for continuous tasks requiring a consistent end-effector trajectory. Before implementing the current version into industrial applications, practical issues such as dynamic collision avoidance, path and trajectory planning, wear and tear of robots and equipment, and interface interoperability must be explored.

7 Verification and validation

The preceding chapters 4-6 presented the design cycles executed for the investigation of the three underlying research questions. The thereby generated results are regarded as research artifacts within the framework of design science research. DSR requires the newly developed artifacts to undergo validation in their application context as part of the relevance cycle. Following the statement of OSMAN BALCI (1997): "V&V is not a phase [...], but a continuous activity throughout the entire life cycle" [OSMA97, p.136], the verification and validation activities were carried out iteratively during the design research as an inherent part of the respective design cycles during the artifact development. While sections 4.5, 5.3, and 6.4 specify the verification and validation efforts to evaluate the respective artifacts, this chapter overviews the applied techniques. Therefore, the following chapter presents a validation in the form of an application to use cases and details the conducted tests aiming at a quantitative verification and validation of the developed research artifacts.

In pursuit of comprehensive validation and investigation of the applicability of the artifacts, established methods and techniques from the existing knowledge base are methodically employed during the verification and validation processes, as presented in section 2.5. In table 7.1 the applied selection of techniques proposed by OSMAN BALCI, RABE ET AL. and SARGENT is shown. The techniques are categorised according to their formality or grade of objectivity [OSMA97]. An unambiguous mapping of a test to the terms verification and validation is not always possible [RABE08, p.13].

Other typical testing methods, like the "Turing test" or "Comparison to other Models" ([RABE08, p.107 ff. SARG10, p.172]), are not applicable to the entirety of the proposed framework of formation planning, since the primary goal of the thesis is not create a novel simulation environment, but rather to provide recommendations of how real systems should behave to demonstrate desired properties. Formal methods, like "Proof of Correctness" or "Predicate Calculus", cannot be applied, since the proposed simulation, artifacts and decision framework are too complex to be formally validated [OSMA97, p.138].

During validation and verification, emphasis is placed on operational validity, as it assesses the general applicability of the results to real-world problems. Nevertheless, data validity, conceptual model validation, and computerised model verification are crucial in determining overall validity.

Submodel testing: Since the framework developed in this thesis consists of artifacts, which can be tested separately, the framework fulfils RABE ET AL.'s conditions for the applicability of submodel (sometimes "unit") testing. In difference to RABE ET AL.'s conditions, the structure of the framework does not reflect the hierarchical structure of a real system since the artifacts build a novel method as decision support, and decisions cannot easily be categorised as hierarchical systems. Nonetheless, the resulting framework is built as a hierarchical system consisting of three central artifacts, which therefore apply to submodel testing. Thus, the

³⁵Formal techniques usually are not applicable to complex simulation models [OSMA97, p.138]. Therefore, no formal techniques were applied here.

Table 7.1: Applied techniques of verification and validation. The techniques are categorised according to formality, as proposed by OSMAN BALCI. *S*, *R* and *B* indicate respectively the source SARGENT, RABE ET AL. and OSMAN BALCI. Moreover, the application of the techniques for conceptual model validation, computerised model verification, and operational validation is given, as well as the section in which the techniques are applied.

Applied	V&V Techniques	Categories				Application stage			Artifact			
		Informal	Static	Dynamic	Formal ³⁵	Conceptual model validation	Computerised model verification	Operational validation	Description model	Executability evaluation	Base placement	Formation planning
✓	Animation ^{S,R,B} (or operational graphics/monitoring ^{S,R,B})	x		x			x			x	x	
✓	Desk checking ^{R,B}	x					x					x
✓	Documentation checking ^B	x					x					x
✓	Structured walkthrough ^{R,B}	x					x					x
✓	Face validity ^{S,R,B}	x				x			x	x	x	x
✓	Fixed value test ^{R,B}		x					x	x			x
✓	Semantic Analysis ^B		x			x			x			
✓	Submodel testing ^{R,B}			x				x	x	x	x	x
✓	Sensitivity analysis ^{R,B}			x				x	x		x	x
✓	Extreme condition test ^{S,R,B}			x				x	x	x		
(✓)	Comparison with other models ^{S,R,B}			x		x				x		
(✓)	Event validity test ^{R,B}			x				x		x		
(✓)	Statistical techniques ^{R,B}			x				x	x	x		
✓	Internal validity ^{S,R,B}			x				x			x	x

individual design artifacts have all been tested, validated and verified in various unit tests, as described in sections 4.5, 5.3, and 6.4. Submodel testing is an essential step in verification, as it identifies flaws and inconsistencies within the programming in its application context.

7.1 Conceptual model validation

To determine whether the artifacts derived in chapters 4, 5, and 6 accurately represent the real-world system of heterogeneous mobile robots in adaptable assembly, with a degree sufficient for planning their formations, the conceptual model requires validation. Consequently, conceptual model validation was conducted on the three artifacts of semantic modelling: the ontology CAPILANO, the executability evaluation by performance maps, and the task clustering for base placement. The validation process involves evaluating the underlying concepts' suitability to increase the level of automation for the formation planning of mobile robots in assembly. The models were validated by the IoP research group during conceptualisation and simulated assembly execution using the face validity principle. Throughout the development process, these artifacts were continuously scrutinised with regard to requirements elicitation, model specification, system and software architecture, the interplay between the implementation of the three artifacts, and causal relationships, as proposed in [VERE21, p.19-34].

Face validity: The newly created ontology CAPILANO (chapter 4) aims to provide a semantic model of mobile robots, components that compose mobile robots, as well as their combinability and dependence on combinatoric derivable capabilities of the composed robot. The criteria used to assess the suitability of CAPILANO (as outlined in table 4.3), such as the identifiability of resources, inheritable capabilities, or the matching of resources to tasks based on capabilities, were demonstrated to be applicable for this specific use case. The domain ontology for CAPILANO, as described in the appendix in chapter 4.3, was continuously adapted and discussed with the assembly experts of the research group "Future Assembly" of the IoP. The findings indicated that both the ontology and the task allocation framework are valid for the specified use case.

The executability evaluation in chapter 5 aims to evaluate, whether a robot equipped with an end-effector is able to execute a given assembly task. During face validation, some deficits in matching the performance metrics to assembly tasks were discussed. For broader application the list of matched metrics has to be extended and the matching itself has to be further tested. Nonetheless, a sufficient degree of conceptual validity for proof-of-concept is apparent, to automate and further specify the executability evaluation. Moreover, the correlation of the introduced average performance metric with analytic executability was tested and proven in section 5.3.

Comparison with other models: The performance map introduced in chapter 5 represents the distribution of performance metrics in the robot workspace, thereby enabling an executability evaluation. Executability hereby refers to extending previous workspace representation by enabling the representation of different performance metrics, to evaluate the executability of specific assembly tasks. As indicated in figure 5.10, the resulting evaluation of reachability of multiple task poses was tested with varying poses, robot models, performance map resolution

and end-effector and compared to the reachability calculation through inverse kinematic. Moreover, the performance map was compared to capability maps during development. Through comparison with other models, the performance map calculation was proven suitable for the application of robot workspace evaluation.

The goal of the base placement and formation planning of chapter 6 is to optimise the task allocation of task poses through task clustering and to optimise the placement of mobile robots whilst integrating the results of the previous research artifacts. To evaluate the validation of the conceptual model, it is necessary to assess the alignment of the underlying task clustering and multistaged heuristics with the actual use case. The assumptions guiding the base placement were discussed in detail with assembly planning experts. Especially the steps of manually limiting the search space for base placements in dependence on the execution direction of the task poses on the product (in front of or behind the product), and the weight matrix to determine which performance metrics should be preferred require reiteration for automation and optimisation. However, the multistaged heuristic is proven to be sufficient for proof-of-concept, and most steps are automated whilst providing evaluation of the executability of formations.

In summation, the conceptual validity of the three developed research artifacts is proven to be sufficient for the intended use.

7.2 Computerised model verification

Computerised model verification necessitates ensuring that the model is correctly implemented, as introduced in section 2.5. Animation, desk checking, documentation checking, and structured walkthrough were conducted to verify that the conceptual model is correctly implemented and the program calculates the executability evaluation, base placement, and formation planning as intended.

Animation: Animation (or operational graphics/monitoring) as verification technique in accordance to RABE ET AL. (2008) and SARGENT (2010) ([RABE08, p.95, 101; SARG10, p.171–172]) regarding the time-dependent behaviour of the formation planning is not possible, since the model does not consider behaviour over time. Instead, the resulting workspace evaluation of chapter 5 can be visualised as a performance map and the planned formation of chapter 6 can be visualised regarding the placement of the physical objects (mobile robot and product) as well as the allocated task poses. Thus animation was applied to the artifacts of chapter 5 and 6. Thereby, the formation planning behaviour is tested for consistency.

Firstly, the correct implementation of the executability evaluation of robot workspaces, as introduced in chapter 5, is verified by animation. The operational behaviour of the modelled performance map and extraction of feasible robot flange poses is displayed graphically. During the implementation of the calculation of the area of feasible robot flange poses, the resulting area was visualised and manually checked for obvious flaws, e.g. deformed areas. One example of the resulting visualisation is shown in figure 5.9. Moreover, the resulting performance map, as a visual representation of the spatial distribution of performance metrics in the robot workspace, is displayed. Python and Matplotlib are utilised to illustrate the performance maps

with different workspace resolutions (Δ), numbers of poses per voxel (N), and the robot models, as shown in figures E.1, E.2, and E.3.

Secondly, the base placement and formation planning, as introduced in chapter 6 were verified through animation. The following intermediate steps were visualised during implementation: The result of the different task pose creation methods, introduced in section 6.3.1, the steps of reducing the search space of possible base placements in section 6.3.2, the resolution of the discrete favoured base placements in the search space in section 6.3.4, and finally, the selected base placements in section 6.3.5. Thereby, Matplotlib is utilised to verify the intermediate results of each implementation step.

Animation and visualisation allow for detecting and resolving scaling errors, displacements, and similar discrepancies. Despite its subjective nature, this verification technique enables the identification and resolution of errors encountered during the implementation process.

Desk checking: The verification and validation method of desk checking was applied during artifact development. The artifacts have been carefully checked for completeness, correctness, consistency and unambiguity as required by BALCI [BALC98, p.356]. The main challenge with this technique is the potential failure to recognise one's own mistakes when revisiting. To address this issue, the strategy was revisited after a period of time whilst reiterating the program during the next research cycle and enhanced with complementary techniques such as documentation checking and structured walkthrough.

Documentation checking: Documentation checking of the central artifacts (corresponding to the results of the underlying research questions) of the formation planning method was conducted for each artifact individually as well as for the entire method. As described in the respective sections regarding verification and validation (sections 4.5, 5.3 and 6.4), the conducted research was documented, the code of the research artifacts and the application ontology, as well as the conducted experiments and their results, were published. The resulting publications are summarised in chapter 9.2. During the publishing process, the artifacts documentation was screened by the reviewers and accepted.

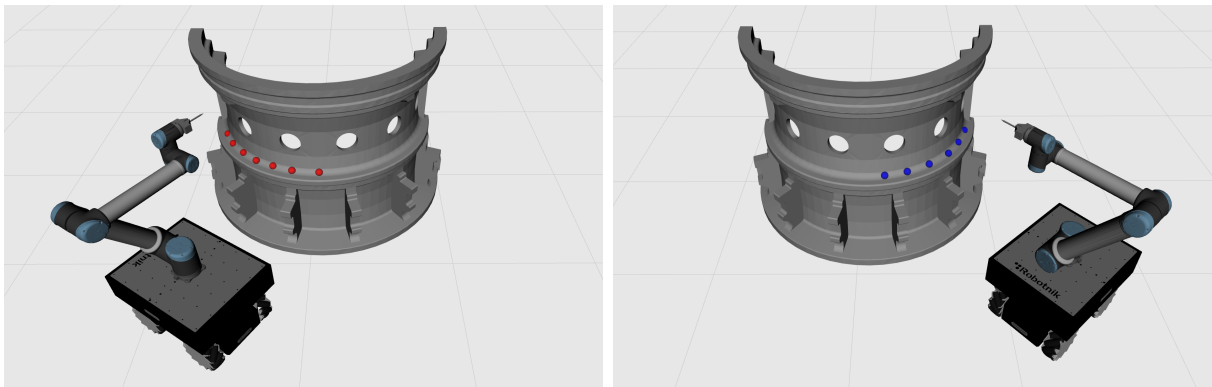
Structured walkthrough: A structured walkthrough of the artifacts (according to RABE ET AL. (2008)) was conducted in internal discussions and interim presentation of results with colleagues at the institute as part of the dissertation process, as well as with colleagues of the research project IoP. Additionally, the main elements of the source code of the artifacts were presented and reviewed as part of the publishing process. Desk checking and documentation checking were repeated during the structured walkthrough of the implemented code, especially during the process of supervising these. The supervised students (see chapter 9.2) were introduced to the existing code base at the beginning of their studies. Therefore, the structured walkthrough was conducted as part of explaining the code base and simultaneously, desk checking and documentation checking were conducted, errors can be spotted while explaining what the code is intended for and looking at the code. Moreover, the documentation was checked, and the students read through it and were told to note questions or remarks related to the documentation.

In conclusion, the computerised model is verified, and the artifacts are correctly programmed and implemented.

7.3 Operational validation

As stated in section 2.5, operational validation determines whether the model's output behaviour is sufficiently accurate for the intended range of applications in dependence on the model's purpose. Therefore, the artifact's behaviour is tested for adherence to the intended purpose in their intended application domain.

The derived artifacts are intended for use in planning formations of mobile robots in assembly stations, ensuring the tasks' executability. Thus, to assess whether the planned formations are indeed executable, the use case scenario of turbine inspection introduced in chapter 6 is first set up in the simulation environment of ROS and then connected to the real world laboratory set up. The scenario consists of one mobile robot and one turbine as the product on which the operations have to be executed. The experiments involved using a UR10 robot by Universal Robotics, which was placed on a mobile base of SUMMIT-XL STEEL mobile robot by Robotnik. The iGPS for Dynamic Tracking and Alignment by Nikon (now 7D Kinematics) was used as the metrology device to measure the goal poses before testing and track the robot's pose during testing (as visualised in the appendix in figures G.1 and G.2). The ROS-based coordination and operation of the robots were carried out using a Dell Precision 7740 with an Intel Core i7-9850H CPU @ 2.60 GHz x 12 and 32 GB memory, running on a 64-bit Ubuntu 18.04.6 LTS operating system. Wireless Local Area Network (WLAN) was used for communication between all the components listed above.



(a) Left base placement in front of the turbine in ROS simulation. (b) Right base placement in front of the turbine in ROS simulation.

Figure 7.1: Validation of base placements in front of a turbine through simulation of the task execution in ROS.

The created use case uses consists of task poses placed on the turbine surface. Figure 7.2 f) shows the turbine with marked goal poses for the end-effector. To execute the tasks, the tip of the end-effector has to reach and mark those task poses. Therefore, the task poses are measured with the iGPS metrology system on the turbine. Before running the practical test, the scenario was run through the autonomous clustering-based task allocation and robot base

placement³⁶ and simulated in ROS. The test scenario as a virtual representation is shown in figure 6.7. Figure 6.7 illustrates the calculated base placements, whilst figure 7.1 displays the simulated base placements and the allocated task poses per base placement. The simulation of the execution of all 13 task poses can be seen in the appendix in figures F.5 and F.6. As can be seen from figures F.5, F.6 and the pictures taken during the practical test (see figure 7.2), all task poses were executable from the assigned base placement.

Sensitivity analysis: Tests with an identical resolution, identical mobile robots, and different task pose distributions were conducted to test the formation planning behaviour regarding reaction to different inputs. Therefore, five test scenarios were derived. The test scenarios and the resulting base placements of the mobile robots are pictured in figure 7.2 and summarised in table 7.2. The analysis illustrates that the number of robots is increased for further distributed task poses, as intended. Moreover, the placement of the robots in relation to the task poses is not optimised in terms of the distance of the base from the poses. This can be attributed to the fact that the distance between the robot base and task poses, condition number, manipulability and dexterity index were weighted equivalently in the score matrix (see equation 6.4 in section 6.3.5). The weighting of the performance metrics in the score matrix has to be further tested and matched to the requirements of different assembly task types to validate executability.

³⁶In this case, a resolution of 150 mm was chosen. This value was determined by iteratively running the test scenario in a loop, gradually decreasing the resolution, and identifying where the final solution score converges.

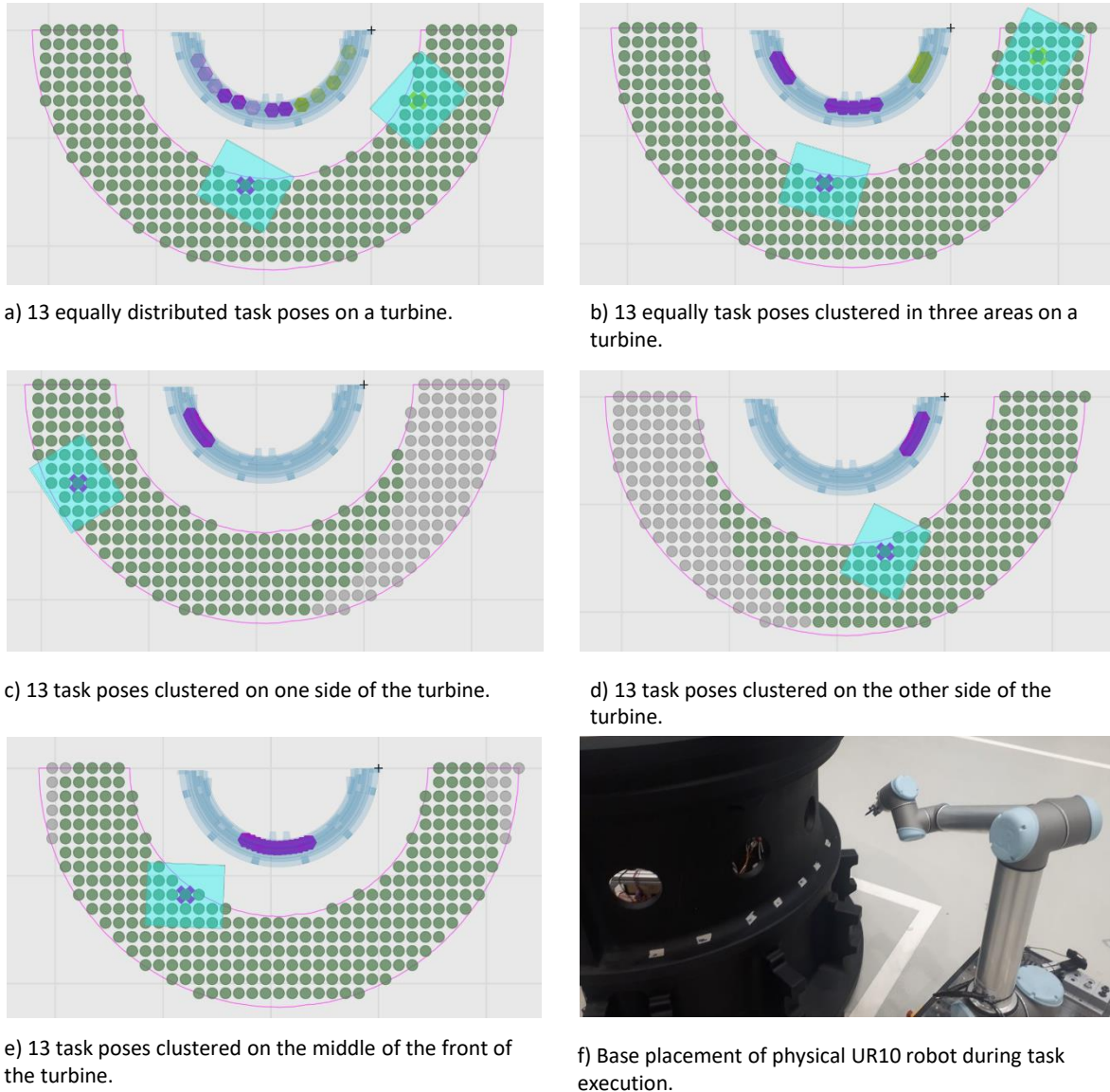


Figure 7.2: Sensivity analysis of the formation planning for inspection task poses on a turbine. Exemplary pictures of the operational validation of base placement and task execution in a test set-up in the application environment at the WZL shopfloor.

Table 7.2: Results of a sensitivity analysis of the formation planning for inspection task poses on a turbine.

Configuration	Computation time [s]	Condition number	Manipulability	Dexterity
a)	216,45	left base placement: 97,4 right base placement: 52,9	2,2 1,38	0,5 0,385
b)	219,048	left base placement: 104,7 right base placement: 52,3	1,86 0,807	0,397 0,224
c)	115,782	75	2,4	0,742
d)	286,782	108,7	2,8	0.691
e)	185,502	85	3,5	0.994

In summary, the operational validity is sufficient. The experiments have shown that the results obtained are suited to real-life scenarios. The primary objective of the developed artifacts was to automate the task allocation and placement of mobile robots in assembly while ensuring the executability of all assembly operations. This was successfully achieved. Especially the time needed for formation planning can be significantly reduced from around 14 weeks in the automotive industry [PELL17, p.98] to several minutes. However, shortcomings were identified in the definition of executability for a broad range of assembly task types (as derived in section 5.3), the optimality of solutions and the assignment of weights to base placement combinations (as derived in section 6.4). Addressing these shortcomings may require iterative refinement of the weight matrix introduced in section 6.3.5. Furthermore, it's crucial for future research to give due consideration to evaluating the feasibility of execution.

7.4 Data validity

The use case data utilised for application and testing procedures of the ontology CAPILANO presented in chapter 4 were thoroughly reflected within the research group "Future Assembly" of the IoP and underwent scrutiny during the publication review process. For the executability evaluation conducted in chapter 5, robot URDF models are taken as input as well as the performance metrics identified in the knowledge base (see section 2.2.2). Each is taken from peer-reviewed and published sources. Therefore, they are valid. To conduct the operational validation of the autonomous clustering-based task allocation and robot base placement artifact, a use case from the laboratory of WZL was utilised.

As intermediate results containing the use cases and, therefore, the utilised data employed for the validation of the three design cycles in the chapters 4, 5 and 6 underwent credibility checking during publication, they are affirmed as credible. During the review of the publication scripts, assembly experts endorsed the credibility of the provided use case data, affirming its fidelity in depicting real-world systems.

7.5 Interim conclusion: Verification and validation

In this chapter, the conducted verification and validation strategy for the individual artifacts of chapters 4 to 6 is summarised, and the conducted verification and validation for the combined artifacts in the enveloping artifact of formation planning is presented. The goal of the validation is to determine whether the designed artifacts answer the initially derived research demand and to provide a systematic basis for an acceptance decision of the entire framework of formation planning. Therefore, an overview of the already conducted verification and validation in sections 4.5, 5.3 and 6.4 is given. Additionally, the formation planning — consisting of task clustering, base placement, and executability evaluation based on capability-based task allocation — is applied to different use cases of turbine inspection.

For all three research areas of the description model and task allocation, executability evaluation and base placement the steps of computerised model verification, conceptual model validation, operational model validation and data validity were conducted in submodel testing. The

developed description model CAPILANO showed sufficient applicability and scalability during the instantiation of the ontology with a specific use case scenario. The initially set competency questions and requirements are met. Therefore, the description model is sufficiently accurate for the intended use. The executability evaluation is validated through animation, face validation, extreme condition testing and compared to existing other models. The correlation of performance metrics average and the analytical solutions of the task poses metric values is shown. The set requirements of representation of the spatial distribution of performance metrics in a robot workspace under consideration of equipped end-effectors are met. Thus, the executability evaluation provides sufficient accuracy. The base placement is applied to assembly use cases, where the results are investigated by testing whether all tasks within a particular cluster can be executed by the designated robot, to evaluate its degree of usefulness for the planning of changeable assembly station formations. The application shows the fulfilment of the set requirements, thereby verifying the applicability to the intended application area.

The scientific discourse conducted ensures the formation planning's conceptual validity. The discourse results are reflected in both the requirements and the added artifacts. The computerised model validation is inherently ensured through compilability and the executability of the code base. To further verify the computer model, informal verification techniques like desk checking or structured walkthroughs are applied. Moreover, the technique of animation is applied to monitor anomalies and identify potential sources of error. The absence of such anomalies is interpreted as an indication of appropriate formation planning. The investigation of operational validity is based on application in a demonstration, multiple simulations and parameter variations for use case scenarios. The application shows appropriate behaviour of the developed formation planning for assembly tasks. In summary, it can be concluded that formation planning provides sufficiently accurate results for initial planning without considering time-dependent behaviour. Assembly system designers are encouraged to perform further simulations of task execution and robot motion.

8 Conclusion and critical reflection

This chapter serves to conclude the research on formation planning by reflecting on the research findings (see section 8.1) and on the methodologies applied during the research (see section 8.2). Clarifying the limitations of the research findings and the implications of the underlying assumptions specifies the applicability of formation planning. Examining the strengths and limitations of the selected methodologies clarifies their impact on the outcomes.

8.1 Reflection on the findings

The research topic of this thesis can be thematically categorised as within the domain of changeable assembly. This thesis contributes to the realisation of operating changeable assembly systems by closing the gap between high-level factory configuration planning and the low-level navigation and control of mobile robots. The primary motivation is to plan the formation of mobile robots in changeable assembly stations, a necessary building block for operating changeable assembly systems.

Formation planning is designed for assembly systems with mobile assembly robots. It can be implemented in various assembly systems to plan mobile robot formations in assembly stations. However, it is intended for and exemplified by LMAS and SPS. The planning of formations is indispensable to realise changeable assembly systems with mobile robots. Formation planning facilitates spatiotemporal positioning, continuous replacement, and reconfiguration of mobile robots. Firstly, it involves identifying a suitable combination of assembly components (such as a robot arm, mobile base, and end-effector) capable of executing all given assembly tasks. Secondly, it encompasses assessing and ensuring the spatial executability of these tasks in a planned formation. A systematic literature review revealed (see chapter 2) that current methodologies for the formation planning of mobile robots in assembly stations and the workspace evaluation of mobile robots are inadequate in evaluating task executability. Notably, the integration of task allocation, workspace analysis, and robot placement had yet to be adequately addressed to use mobile robots' capabilities in assembly operations fully. Based on the identified requirements and research gaps, the primary research question is formulated as follows: ***Can capability-based formation planning of mobile assembly robots in stations of changeable assembly systems ensure the executability of the assembly tasks?***

This primary research question is decomposed into three underlying research questions to investigate the description model, the executability evaluation and the final base placement and formation planning.

URQ1 Description model: Answering the first underlying research question URQ1 resulted in two central building blocks, as summarised in section 4.6: First, the **semantic description model of mobile robots and their capabilities**, and second, the **capability-based allocation of tasks** to combined mobile robots. A formal description of combinable components (mobile base, robot arm and end-effector) to build mobile robots was provided, ascribing the resulting capabilities of the composed mobile robot. This allows the selection and combination of

components into one composed mobile robot with the semantic capabilities to execute the tasks as a basis for the subsequent evaluation of practical executability. Therefore, **capability-based executability** is ensured by ascribing capabilities to the composed robot in dependence on the equipped and combined components and allocating the tasks to robots accordingly.

However, the creation of a description model necessitates the abstraction of realistic environmental conditions and the formulation of assumptions, as the real world can never be exactly modelled. The description model relies on manual modelling of component compatibility, which requires expertise in the software and hardware interfaces of multiple robot and equipment vendors. In addition, incorporating executability through semantic capabilities overlooks the state of the resources, such as wear and tear or battery status, which affects their usability.

URQ2 Executability evaluation: Beyond a description, the operation of changeable assembly stations requires an answer to the second underlying research question URQ2 regarding the **evaluation of the executability of assembly tasks in the workspace of a mobile robot under consideration of equipped end-effectors**. The resulting building blocks and findings are summarised in section 5.4. The **performance map** represents the spatial distribution of performance metrics — evaluating different robot capabilities and, therefore, the executability of tasks — in the robot's workspace. The developed representation in the performance maps represents the valid workspace of the combined resource, resulting from the simultaneous consideration of the robot and **equipped end-effector**. The performance map incorporates different performance metrics depending on the assembly task. Performance metrics are matched to evaluate the executability of specific assembly tasks. Therefore, the **spatial executability** of a task pose allocated to a robot at a specific base placement can be evaluated.

The influence of equipped end-effectors continues to depend on manual input to assess the degree of functional redundancy. This assessment varies according to both the specific assembly task and the type of end-effector used.

For a broader application field, the representation of the valid robot workspace in a performance map should be extended to include different robot topologies. Currently, the performance map is limited to open kinematic chain 6-axis robots. While this robot type is very common, hyperredundant (more than 6-axis) robots are increasingly used in industry.

In addition, the current implementation offers a limited range of considered performance metrics and assembly tasks. Whilst demonstrating versatility in different contexts, practical application requires a wider range of tasks and metrics to adequately address the diverse requirements inherent in assembly processes.

The argument that a high manipulability measure guarantees the dynamic execution of all assigned task poses necessitates further investigation in practical use cases and industrial applications, especially for continuous tasks demanding a consistent end-effector trajectory.

URQ3 Base placement and formation planning: An autonomous **clustering based task allocation and robot base placement** has been developed based on capability-based task allocation and end-effector including workspace evaluation, in the context of the third

underlying research question URQ3. The central result is a multistage method that groups distributed assembly task poses into multiple clusters. Each cluster is assigned to a single robot from a single base placement for execution. The method minimises the number of base placements and assigns all task poses to one mobile robot at one base placement. The resulting **formation is evaluated for executability** by applying the previously defined performance metrics represented in performance maps. Incorporating different robot models, task types and product models as input, the autonomous allocation and base placement are adaptable and suitable for changing assembly scenarios.

To improve on the computation time and solution optimality, an enhanced implementation could reduce the complexity and computational time required to solve the base placement problem. Instead of the multistaged heuristic, an optimisation problem could be formulated, and established solving strategies could be employed and evaluated based on criteria such as computation time and solution optimality. The current implementation should be considered a prototype.

Some steps in the multistage heuristic still require manual input (e.g., excluding possible base placements behind the product). These steps should be automated in the future to reduce the manual effort and achieve a higher degree of automation.

The presented formation planning method reduces the effort of planning the task allocation and base placement of robots in assembly stations from previously 14 weeks of manual planning time [PELL17, p.98] to less than a day — including modelling of the resources in the ontology and running the formation planning framework of task allocation, executability evaluation and base placement.

8.2 Reflection on the research methodology

Design Science Research was selected as the foundational research framework for this thesis. The iterative nature of the research methodology — embodied by the three iterative DSR cycles relevance, rigour, and design — enforces constant evaluation and adaption of the research artifacts. The continuous evaluation of partial results allows for early identification of errors, and constant improvement is implemented. The application of the iteration cycles inherent to DSR ensures the integration of findings from the previous research cycle, enables adapting to findings from subsequent research artifacts and warrants continuous verification and validation. Whilst following the cyclic approach is beneficial during conduction, representing the respective iteration cycles and intermediate results is not helpful for documentation of the research findings in written form and publications, as the content would repeat itself. Therefore, only an overview of the major design cycles conducted while answering the underlying research question and the obtained intermediate improvements and results is given at the beginning of chapters 4-6.

Incorporating the knowledge base within the rigour cycle unveiled applicable foundations essential for crafting artifacts corresponding to each underlying research question. Multiple iterations of the design cycles were conducted to develop, improve and assess the semantic modelling, executability evaluation and base placement of the mobile robots in assembly. As per critical reflection in section 8.1 and the previous publication of partial research results, the

artifacts are added to the knowledge base. The application of the design and rigour circles benefited the research process as methodological approaches. The design and rigour cycles were conducted repeatedly. The step of feedback of the developed artifacts from the design cycle into the environment via the relevance cycle is viewed more critically. The lack of industrial deployment of lineless and changeable assembly systems results in a lack of real-world use cases to apply the developed method in practical and realistic application scenarios. Therefore, the pending step is practical application in the industry and the iteration and adaption of the artifacts based on the feedback from the industry.

Reflecting on the applied verification and validation methods and techniques, one must be aware that “testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence” [NBUX69, p.16]. In other words, validation can never objectively ensure the applicability of the model [SARG96, p.56; RABE08, p.2]. Various methods and techniques with varying levels of objectivity were utilised to perform verification and validation during those stages. The techniques were selected to encompass a wide spectrum of objectivity. thereby, the selection of techniques was subjectively, as neither SARGENT (2010), RABE ET AL. (2008), nor OSMAN BALCI (1997) suggest a systematic approach to choosing fitting techniques.

In conclusion, applying the iterative DSR as a research methodology was a successful and helpful approach to structuring the research process. Moreover, selecting verification and validation techniques listed by SARGENT (2010), RABE ET AL. (2008), and OSMAN BALCI (1997) provided a guideline for multiple

9 Summary and outlook

This final chapter summarises the thesis. It presents the contents of each chapter, starting with the motivation and business needs, followed by the knowledge base, the developed artifacts, and their application and validation. The research artifacts are mapped to the initially identified research deficits, and the artifacts are allocated into the levels of the IoP. The insights and results from the thesis are overviewed, and prospects for future adaptation and extension of the results are outlined.

9.1 Summary

Manufacturers must enhance the resilience of changeable assembly systems against external disturbances, such as a shortage of skilled manual workforce or disruption in the supply chain. The increased flexibility and changeability of assembly systems have increased the complexity and requisite effort in assembly planning. Manual or time-consuming planning for personalised products with single or small lot sizes cannot be amortised through reuse as it can be for products with larger lot sizes. Consequently, the business need arises for automated and cost-effective formation planning, including task allocation and base placement of robots in assembly.

Formation planning, as a novel method to place mobile robots in assembly stations under consideration of capabilities and spatial distribution of executability metrics in the robot's workspace, is developed and investigated in this thesis to address these needs. In summation, the planning time for the spatial placement of mobile robots in formations in assembly stations is reduced. The proposed formation planning method provides formations that ensure the executability of the given assembly tasks. The planning process is automated, potentially reducing the manual effort and workload in changeable assembly.

Figure 9.1 overviews the developed research results and artifacts and relates them to the previously identified research deficits and three research areas, as presented in figure 3.3. As a first research result, a **description model** of mobile assembly robots, their components and resulting capabilities in the form of an ontology called CAPabLIty-based resource AllocationN Ontology is derived, and a subsequent capability-based task allocation is devised. Two research results contribute to the **executability evaluation**: First, performance metrics are matched to task types to provide the means to evaluate the capabilities required to perform assembly tasks, resulting in performance maps representing the spatial distribution of performance metrics in the robot workspace. Second, end-effectors are integrated into the evaluation of executability, by calculating the area of feasible robot flange poses, in dependence on the combination of end-effector and assembly task type. The third research area, the **base placement and formation planning**, integrates the previous research results. Formations of mobile robots as allocation of task poses to robots located at defined base placements are planned, ensuring the executability of the allocated assembly tasks. The formation planning presented here can be categorised between factory configuration planning and the process execution and control of

individual resources, thereby closing the gap between high-level factory planning and low-level robot control.

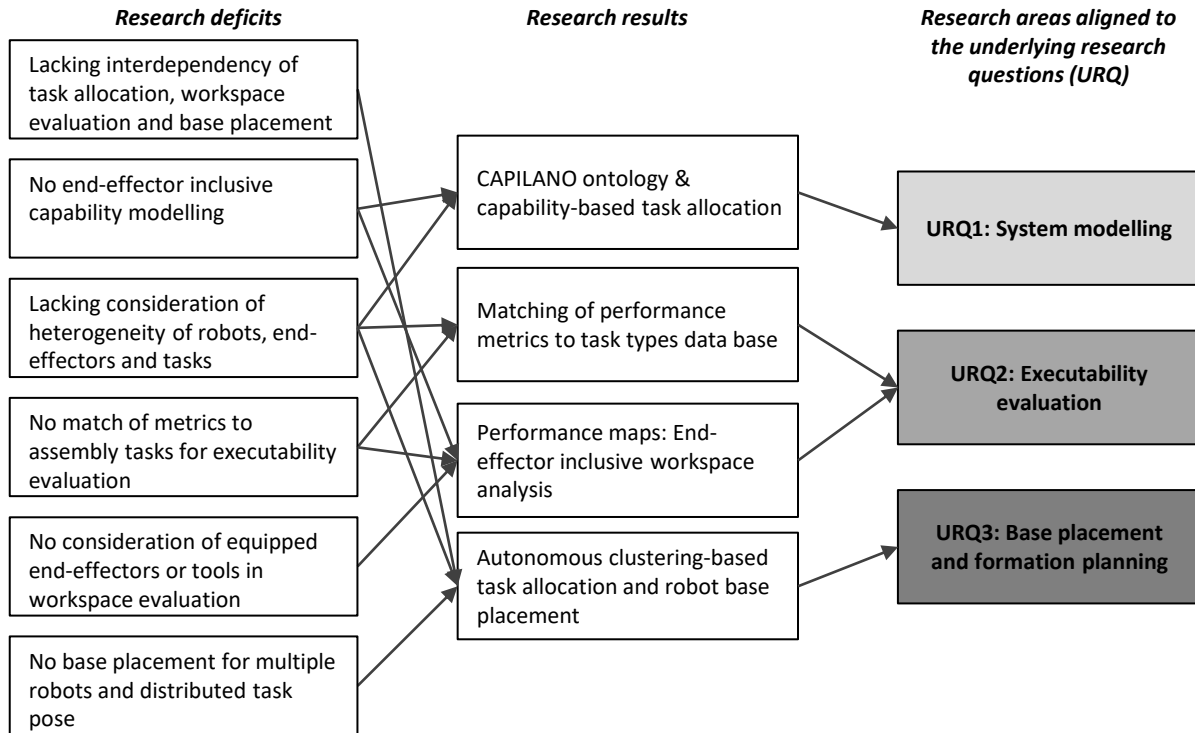


Figure 9.1: Mapping the developed research results to the previously identified research deficits and research areas.

As introduced in figure 1.2 and section 1.2 the research conducted is closely related to the research endeavours of the IoP. Figure 9.2 places the developed artifacts and building blocks into the levels of the framework representation of the IoP whilst highlighting the allocation to the three research areas of this thesis. Based on the input data (e.g. semantic resource, end-effectors, robots or process charts) of the shopfloor level, the digital twins and analytics level constitutes the ontology CAPILANO, performance metrics, end-effector incision, robot base placement and task clustering, analysing the accessible data. The analysis results are transferred into a capability-based task allocation, executability evaluation and finally the station formation on the decision support level.

Chapter 1 provides the practical motivation and heuristic framework of the scientific process. The problem statement and business need to automate the formation planning of mobile robots in assembly stations are delineated. The goal to close the gap between macroscopic factory planning and microscopic robot control by the mesoscopic level of formation planning is postulated.

In chapter 2, the foundation and knowledge base of the research areas are summarised. Methods of robot workspace evaluation, existing performance metrics and their applications, as well as established ontologies in the manufacturing domain are overviewed. The need for a systematic literature review on formation planning is identified.

Chapter 3 presents a systematic literature review, exploring the combination of the research areas of robot modelling, workspace evaluation, and base placement needed for formation

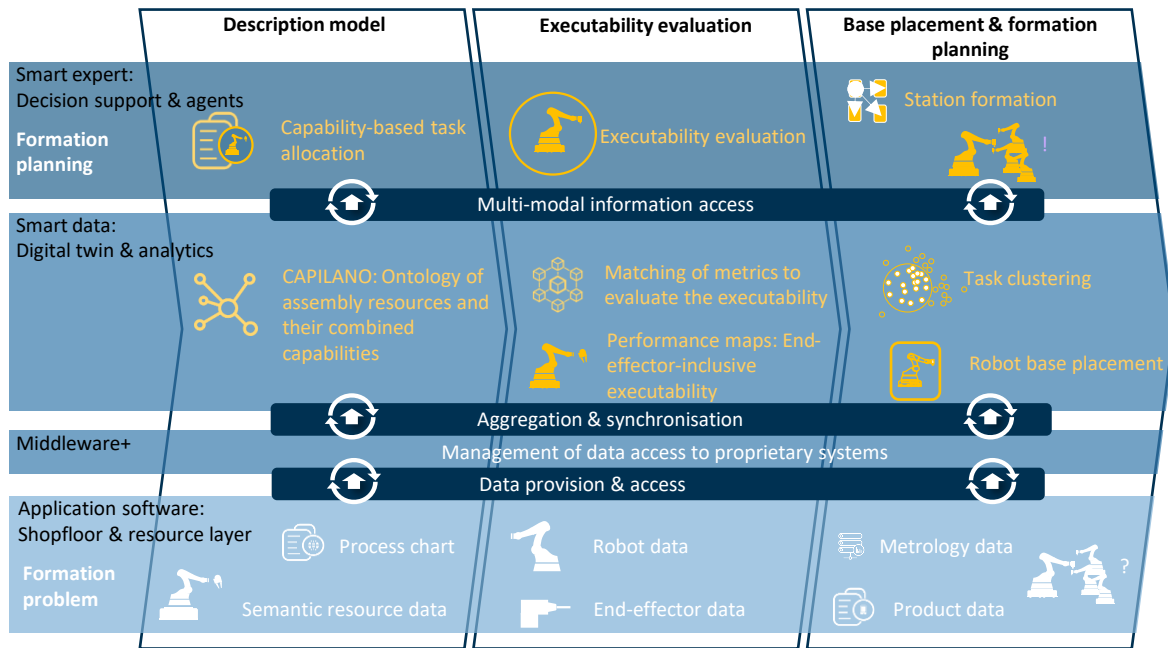


Figure 9.2: Allocation of the developed research results (in orange) into the levels of the Internet of Production (based on [BREC17]).

planning. The review raises research deficits, as shown in figure 9.1. The primary research question, as well as the underlying research questions, are formulated.

In chapter 4 the semantic and syntactic description of product, process, resources and capabilities in an assembly station is conducted. Therefore, the focus is on the representation and inheritance of capabilities and the combination of resources, consisting of mobile bases, robot arms, and end-effectors. A subsequent, capability-based task allocation framework is presented, enabling the choice of a combination of resources fitting to execute given assembly tasks.

The end-effector inclusive executability evaluation is introduced in chapter 5. By considering the geometric dimensions and implication of the degree of redundancy for task execution of equipped end-effectors on robots, the executability evaluation is extended. Performance metrics are chosen to evaluate a robot's ability to execute different assembly tasks. The research results are summarised in the representation of performance metric distribution in robot workspaces in performance maps.

In chapter 6 a clustering-based task allocation and base placement method is introduced, integrating the previous research results into one formation planning approach. The multistaged heuristic successively reduces the area of possible base placement, in dependence on the dimensions of the product, the robot arm and the mobile base, as well as the performance map. Finally, the preselected possible base placements are evaluated with different performance maps to find a combination of base placements, allowing for the successful execution of all tasks whilst minimising the number of base placements.

Whilst partial verification and validation of the respective research results is conducted in the three main chapters 4-6, chapter 7 presents the final verification and validation of the entire formation planning. The artifacts are validated based on the methods and techniques of

SARGENT (2010), RABE ET AL. (2008), and OSMAN BALCI (1997). The investigation is based on an application scenario pertaining to the assembly of truck frames. The application and variation of parameters show plausible behaviour for the executability evaluation and formation planning of multiple mobile robots. The formation planning exhibited considerable potential for automating and optimising the planning process associated with the configuration, task allocation, and base placement of mobile robots within changeable assembly stations.

Chapter 8 concludes the research process and the research findings. Limitations of the research findings are specified, and the applied research methodology is critically reflected. The research questions are answered with a sufficient degree of confidence. Nevertheless, each research result is subject to simplifying assumptions, limitations and constraints in technical feasibility and applicability as explained in sections 4.6, 5.4, 5.4 and summarised in 8.1. Therefore, the subsequent section 9.2 highlights future research potential to overcome these limitations and extend the industrial applicability of the obtained artifacts.

Finally, a reflection is given on the level of achievement in addressing both the primary research question and the three underlying research questions:

1st underlying research question: *How can multipurpose assembly robots, their capabilities and assembly tasks be modelled for a task allocation?* ✓

Answer: The ontology CAPILANO models assembly robots as composed of robot arms, mobile bases and end-effectors and ascribes capabilities in dependence on combined components. The adjacent task allocation allocates tasks in dependence on capabilities.

2nd underlying research question: *Can the executability evaluation of assembly tasks be extended whilst considering equipped end-effectors on mobile robots?* (✓)

Answer: Yes, by determining feasible robot flange poses from the degree of functional redundancy depending on the combination of end-effector and assembly task type, equipped end-effectors can be considered in the executability evaluation. Task executability is determined by assigning and calculating performance metrics in the robot workspace and representing the distribution of these metrics in performance maps.

3rd underlying research question: *How can assembly tasks be clustered to plan formations with reduced base placements of mobile assembly robots?* ✓

Answer: A multistaged heuristic applies K-Means and OPTICS clustering of task poses and successively reduces the number of potential base placements to finally identify a minimised number of base placements allowing execution of assigned task clusters.

Primary research question: *Can capability-based formation planning of mobile assembly robots in stations of changeable assembly systems ensure the executability of the assembly tasks?* (✓)

Answer: Yes, the combination of capability-based task allocation, robot workspace analysis and base placement enables the planning of executable formations.

The research results and artefacts effectively address the first and third underlying research questions regarding robot and capability modelling and base placement, respectively. However, the second underlying research question is partially addressed. In the executability evaluation,

the impact of equipped end-effectors still relies on manual input to determine the degree of functional redundancy, which varies depending on the assembly task and end-effector type. Furthermore, the current implementation is restricted to 6-axis robots with limited performance metrics and assembly tasks. While demonstrating general applicability, practical implementation requires a broader range of tasks and metrics to address the diverse requirements of assembly processes. Therefore, the primary research question can only be partially answered positively as well. While the combination of executability evaluation and associated base placement increases the likelihood of executability, ensuring executability requires simulation of planned base placement and task allocation.

9.2 Future research and outlook

The obtained research results open up new perspectives for future research projects, deepening the understanding of the research areas in formation planning as well as in the context of the practical implementation of changeable assembly systems, necessitating the integration of formation planning into a framework, considering factory planning and robot control. Therefore, the outlook focuses on the refinement of the artifacts in themselves to enhance their technology readiness level and the integration of the artifacts in the intended application field of changeable assembly.

Artifact refinement

Description model refinement: The automated derivation of URDF models from the provided robot and end-effector dimensions and topology to directly simulate the described assembly station could accelerate the planning and simulation process. Moreover, the modelling of the robot and peripheral components like mobile bases and end-effectors can be extended by the compatibility of hardware and software interfaces of the physical components. Hardware and software compatibility must be manually implemented and integrated explicitly into the ontology. Investigating the compatibility between supplier interfaces and various types of robots and end-effectors would significantly enhance practical applicability.

Extending the executability evaluation: The executability evaluation could be expanded by including additional metrics and a broader range of assembly tasks in the proposed matching of metrics to tasks and performance map creation. Improvement of the executability is expected for further investigation of the weighting of the metrics for different task types to select preferred robot base placements. It is expected that an explanation of the performance metric and options for applying metrics would provide users with a deeper understanding of the metrics without necessitating previous expert knowledge. Additionally, the current calculation of performance maps could be extended to consider different robot topologies, such as robots with more than six degrees of freedom or delta robots with a closed kinematic chain.

Moreover, the interdependencies of the process, the task poses, and the executing robot have to be considered. Currently, the temporal dimension of task execution is not considered. Still, it does impact task executability — if continuous trajectories over a long distance are required — as well as the order of allocated tasks during task allocation. Regarding task allocation, the

current task clustering and base placement could be applied as the input for a travelling salesman problem, solving for the required characteristic of minimal execution time, quality or the like.

Alternative solution to the multistaged formation planning heuristic: The current representation of metrics and constraints to express and determine feasible and executable formations of mobile robots in changeable assembly could be transferred into an optimisation problem formulation. This would provide the opportunity to compare different solvers to derive optimised solutions for formation planning. Potentially, established solvers might be suitable to solve the problem faster than the proposed heuristic and provide optimal solutions.

Integration into a digital twin: The formation planning can be expanded by facilitating the linkage between the ontological model, the simulation environment and their corresponding physical instances in the real world. This integration enables the creation of a digital shadow — enabling a real-time representation of the real world in the modelling and simulation environment — as a first step and a digital twin as a second step. A digital twin closes the loop between modelling and simulation and the real world by enabling the control and feedback from the virtual representation to the real world. The digital twin enables real-time decisions and reactions based on the modelled and simulated process behaviour.

Application and integration into the operation of changeable assembly system

The practical significance of the developed research artifacts relies upon the premise that the application of mobile robots in assembly will increase and demonstrate to be beneficial and efficient. Therefore, enhancements in assembly planning, robot navigation, and control must be combined with the presented findings here. Thus, this thesis should not be viewed as stand-alone. Rather, it must be seen as a partial advancement towards increasing the technology readiness level of mobile robots for changeable assembly systems.

The current formation planning does not consider potential challenges emerging during the practical integration into the industrial assembly. Integrating the mesoscopic level of formation planning between the macroscopic level of factory planning and the microscopic level of robot control necessitates the design of an overarching planning architecture or the integration into an existing one. The transfer of data between the three levels and the potential re-iteration of decisions dependent on the feedback from either other level must be considered. Before deploying the developed formation planning in industrial applications, practical considerations, including path and trajectory planning, dynamic collision avoidance, wear and tear management of robots and equipment, and interface interoperability, demand a thorough examination. Moreover, compatibility with the prevalent factory planning and the subsequent execution and control of robots has to be ensured, and a communication architecture between the resources and decision agents has to be created.

The subsequent assumptions were formulated and must be verified for their applicability before the formation planning method can be utilised in industrial settings: Firstly, the formation planning and robot placement assume a factory without human operators or a human workforce. Generally, humans could be modelled as resources in the ontology and considered in the task allocation and base placement, but since the goal is to automate assembly, they are not considered here. Secondly, a suitable environment is assumed. This necessitates a level and

stable surface, which is necessary for wheeled mobile robots. In addition, sufficient metrological coverage is a prerequisite for localisation and navigation. The listed assumptions must be met and accounted for for practical applicability.

The evaluation of assembly station formations solely incorporates feasibility studies and does not reflect a business viewpoint or performance metrics indicating typical operations research metrics. Before the created research artifacts can be applied in the industry, additional evaluation criteria, like performance, overhead costs, throughput, station footprint, investment, production quality and alike, should be considered. [LEIT08, p.4; ELMA09, p.10; FECH18, p.41] Quality, cost, and throughput time are relevant decision criteria for industrial applications. Therefore, they are among the most considered criteria in operations research and applied criteria in optimisation problems for designing assembly stations. The influence of different formations on quality, cost and throughput time has to be studied. A formalised description of the correlation between formations and operational metrics can be included as selection criteria in the last assessment step of the autonomous clustering-based task allocation and robot base placement.

Once the formation planning step has been completed, the planned formation must be supplemented by complex path and trajectory planning algorithms that incorporate dynamic collision avoidance. Moreover, the communication, control, and regulation of individual resources has to be designed, integrated and verified. Applying multi-robot path and trajectory planning and whole-body motion planning, taking the planned formations as input, is expected to yield relevant insights. The application and combination of formation planning with whole-body motion planning of mobile robots necessitate 5G-based wireless communication and real-time cloud-to-robot connectivity, which requires an edge computing architecture.

References

- [22a] MANPOWERGROUP EMPLOYMENT OUTLOOK SURVEY. 2022. URL: https://go.manpowergroup.com/hubfs/MEOS/2022_Q4/Global_EN_MEOS_Report_4Q22.pdf (visited on 2022).
- [22b] Over 4 million mobile robots will be installed by the end of 2027. 2022. URL: <https://www.controlengurope.com/article/194959/Over-4-million-mobile-robots-will-be-installed-by-the-end-of-2027.aspx> (visited on 11/01/2023).
- [23a] iGPS 7DoF Tracking Technology. 2023. URL: <https://7dkmetrology.com> (visited on 10/30/2023).
- [23b] What does "Relevance" mean in Scopus? 2023. URL: https://service.elsevier.com/app/answers/detail/a_id/14182/supporthub/scopus/track/BPNlyQpQDv8C~Qu~Gm4a~yCcd0Eq2S75Mv~P~zj~PP8G/ (visited on 06/09/2023).
- [ABDE06] K. Abdel-Malek, J. Yang, D. Blackmore, and K. Joy. "Swept volumes: fundation, perspectives, and applications". In: *International Journal of Shape Modeling* 12.01 (2006), pp. 87–127.
- [ABDO17] M. Abdolrazzagah-Nezhad and Salwani Abdullah. *Job Shop Scheduling: Classification, Constraints And Objective Functions*. 2017. DOI: 10.5281/zenodo.1129630.
- [AKIN18] I. Akinola, J. Varley, B. Chen, and P. K. Allen. "Workspace aware online grasp planning". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 2917–2924.
- [AMEL13] J. Amelingmeyer. *Wissensmanagement: Analyse und Gestaltung der Wissensbasis von Unternehmen*. Springer-Verlag, 2013. ISBN: 332281730X.
- [ANGE16] J. Angeles and F. C. Park. "Design and performance evaluation". In: *Springer Handbook of Robotics*. Springer, 2016, pp. 399–418.
- [ANGE92] J. Angeles and C. S. López-Cajún. "Kinematic isotropy and the conditioning index of serial robotic manipulators". In: *The international journal of Robotics Research* 11.6 (1992), pp. 560–571.
- [ANGR20] L. S. Angreani, A. Vijaya, and H. Wicaksono. "Systematic literature review of industry 4.0 maturity model for manufacturing and logistics sectors". In: *Procedia manufacturing* 52 (2020), pp. 337–343.
- [ANKE99] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. "OPTICS: Ordering points to identify the clustering structure". In: *ACM Sigmod record* 28.2 (1999), pp. 49–60.
- [AREN22] J. Arents and M. Greitans. "Smart industrial robot control trends, challenges and opportunities within manufacturing". In: *Applied Sciences* 12.2 (2022), p. 937.

- [ASAD16] N. Asadi, M. Jackson, and A. Fundin. “Drivers of complexity in a flexible assembly system-A case study”. In: *Procedia CIRP* 41 (2016), pp. 189–194.
- [AVHA24] A. Avhad, C. Schou, A. Kluge-Wilkes, A. Göppert, R. H. Schmitt, and O. Madsen. “A Reference Architecture for Swarm Production: A Top-down Approach Supported by Holonic ARTI”. In: *Journal of Industrial Information Integration* (2024). (In review).
- [AZAD19] K. Azadeh, R. De Koster, and D. Roy. “Robotized and automated warehouse systems: Review and recent developments”. In: *Transportation Science* 53.4 (2019), pp. 917–945.
- [BAAD03] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, and D. Nardi. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003. ISBN: 0521781760.
- [BADE19] S. R. Bader, M. Maleshkova, and S. Lohmann. “Structuring reference architectures for the industrial internet of things”. In: *Future Internet* 11.7 (2019), p. 151.
- [BALC98] O. Balci. “Verification, validation, and testing”. In: *Handbook of simulation* 10.8 (1998), pp. 335–393.
- [BANA15] W. Banas, A. Sekala, A. Gwiazda, K. Foit, P. Hryniewicz, and G. Kost. “Determination of the robot location in a workcell of a flexible production line”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 95. 1. IOP Publishing. 2015, p. 012105.
- [BAUE20] T. Bauernhansl and R. Miede. “Industrielle Produktion—Historie, Treiber und Ausblick”. In: *Fabrikbetriebslehre 1: Management in der Produktion* (2020), pp. 1–33.
- [BEER14] J. M. Beer, A. D. Fisk, and W. A. Rogers. “Toward a framework for levels of robot autonomy in human-robot interaction”. In: *Journal of human-robot interaction* 3.2 (2014), pp. 74–99. ISSN: 2163-0364. DOI: 10.5898/JHRI.3.2.Beer.
- [BEES15] P. Beeson and B. Ames. “TRAC-IK: An open-source library for improved solving of generic inverse kinematics”. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2015, pp. 928–935.
- [BEES21] P. Beeson, S. Pfeiffer, S. Hart, M. Günther, M. Lanighan, A. Huaman Quispe, and K. Brameld. *trac_ik. Public repository for the current stable version of the TRAC Labs’ IK solver*. Accessed = 2023-11-02. 2021.
- [BEET18] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels. “Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 512–519.

- [BEHE23] M. Behery, P. Brauner, A. Kluge-Wilkes, R. Baier, A. Mertens, R. H. Schmitt, M. Ziefle, and G. Lakemeyer. "Digital Shadows for Robotic Assembly in the World Wide Lab". In: *Procedia CIRP* 120 (2023). 56th CIRP International Conference on Manufacturing Systems 2023, pp. 165–170. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2023.08.030>.
- [BEIE19] C. Beierle and G. Kern-Isberner. *Methoden wissensbasierter Systeme*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019. ISBN: 978-3-658-27083-4. DOI: 10.1007/978-3-658-27084-1.
- [BESSL18] D. Beßler, M. Pomarlan, and M. Beetz. "Owl-enabled assembly planning for robotic agents". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2018, pp. 1684–1692.
- [BISH06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [BODE21] M. Bodenbenner, M. P. Sanders, B. Montavon, and R. H. Schmitt. "Domain-specific language for sensors in the internet of production". In: *Production at the leading edge of technology: Proceedings of the 10th Congress of the German Academic Association for Production Technology (WGP), Dresden, 23-24 September 2020*. Springer. 2021, pp. 448–456.
- [BØGH12] S. Bøgh, O. S. Nielsen, M. R. Pedersen, V. Krüger, and O. Madsen. "Does your robot have skills?" In: *Proceedings of the 43rd international symposium on robotics*. VDE Verlag GmbH. 2012.
- [BONE01] I. A. Bonev and J. Ryu. "A geometrical method for computing the constant-orientation workspace of 6-PPRS parallel manipulators". In: *Mechanism and machine theory* 36.1 (2001), pp. 1–13.
- [BRAN20] M. Brandmeier. "Ganzheitliches ontologiebasiertes Wissensmanagement im Umfeld der industriellen Produktion". PhD thesis. FAU University Press, 2020. DOI: 10.25593/978-3-96147-276-5.
- [BREC17] C. Brecher, F. Klocke, R. Schmitt, and G. Schuh. *Internet of Production für agile Unternehmen: AWK Aachener Werkzeugmaschinen-Kolloquium, 18. bis 19. Mai 2017*. Apprimus Verlag, 2017.
- [BROW84] J. Browne, K. Rathmill, S. P. Sethie, and K. E. Stecké. "Classification of flexible manufacturing systems". In: *FMS Magazine* (1984), pp. 114–117.
- [BUCK19a] A. F. Buckhorst, A. Kluge-Wilkes, and R. H. Schmitt. "Flying Metrology and Defect Identification for Aircraft Surface Inspection: Automated optical investigation of aircraft surfaces after lightning strike events". In: *PhotonicsViews* 16.1 (2019), pp. 68–71. DOI: <https://doi.org/10.1002/phvs.201900009>.
- [BUCK19b] A. F. Buckhorst, G. Huettemann, L. Grahn, and R. H. Schmitt. "Assignment, Sequencing and Location Planning in Line-less Mobile Assembly Systems". In: *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*. Vol. 57. 2019, pp. 227–238. DOI: 10.1007/978-3-662-59317-2₂₃.

- [BUCK22] A. Buckhorst, M. do Canto, R. Rabelo, and R. Schmitt. "A Holonic Control System Approach for Line-less Mobile Assembly System Operations". In: *Procedia CIRP* 107 (2022). Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022, pp. 1449–1454. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2022.05.173>.
- [BUCK23] A. F. Buckhorst. "Planung und Steuerung von Abläufen in linienlosen mobilen Montagesystemen; 1. Auflage". Druckausgabe 2023. – Auch veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2023. Dissertation. Aachen: RWTH Aachen University, 2023, 1 Online-Ressource : Illustrationen, Diagramme. DOI: 10.18154/RWTH-2023-09029.
- [BURG15] F. Burget and M. Bennewitz. "Stance selection for humanoid grasping tasks by inverse reachability maps". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 5669–5674. DOI: 10.1109/ICRA.2015.7139993.
- [BUTT20] J. Butt. "A strategic roadmap for the manufacturing industry to implement industry 4.0". In: *Designs* 4.2 (2020), p. 11.
- [CAO19] Q. Cao, C. Zanni-Merk, and C. Reich. "Ontologies for Manufacturing Process Modeling: A Survey". In: *Sustainable Design and Manufacturing 2018*. Ed. by D. Dao, R. J. Howlett, R. Setchi, and L. Vlacic. Vol. 130. Smart Innovation, Systems and Technologies. Cham: Springer International Publishing, 2019, pp. 61–70. ISBN: 978-3-030-04289-9. DOI: 10.1007/978-3-030-04290-5{\textunderscore}7.
- [CAST08] G. Castelli, E. Ottaviano, and M. Ceccarelli. "A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators". In: *Mechanics based design of structures and machines* 36.1 (2008), pp. 14–33.
- [CASW] T. A. Caswell, M. Droettboom, A. Lee, and Iliott Sales de Andrade et al. *matplotlib/matplotlib: REL: v3.3.4*. Version v3.3.4. DOI: 10.5281/zenodo.4475376.
- [CHAN88] P. H. Chang. "A Dexterity Measure for the Kinematic Control of Robot Manipulator with Redundancy". In: (1988).
- [CHEN07] C. Chen and J. Angeles. "Generalized transmission index and transmission quality for spatial linkages". In: *Mechanism and Machine Theory* 42.9 (2007), pp. 1225–1237.
- [CHEN16] W. Chen, Y. Tang, and Q. Zhao. "A novel trajectory planning scheme for spray painting robot with Bézier curves". In: *2016 Chinese Control and Decision Conference (CCDC)*. IEEE. 2016, pp. 6746–6750.
- [CHEN19a] W. Chen, S. Zhou, Z. Pan, H. Zheng, and Y. Liu. "Mapless collaborative navigation for a multi-robot system based on the deep reinforcement learning". In: *Applied Sciences* 9.20 (2019), p. 4198.

- [CHEN19b] Y. Cheng, F. Sun, Y. Zhang, and F. Tao. "Task allocation in manufacturing: A review". In: *Journal of Industrial Information Integration* 15 (2019), pp. 207–218.
- [CHEN22] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai. "A review of visual SLAM methods for autonomous driving vehicles". In: *Engineering Applications of Artificial Intelligence* 114 (2022), p. 104992. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2022.104992>.
- [CHIU88] S. L. Chiu. "Task compatibility of manipulator postures". In: *The international journal of robotics research* 7.5 (1988), pp. 13–21.
- [CHOW21] P. Chowdhury, S. K. Paul, S. Kaisar, and M. A. Moktadir. "COVID-19 pandemic related supply chain studies: A systematic review". In: *Transportation research. Part E, Logistics and transportation review* 148 (2021), p. 102271. DOI: 10.1016/j.tre.2021.102271.
- [CIRP20] P. CIRP. "Wörterbuch der Fertigungstechnik Bd. 4/Dictionary of Production Engineering Vol. 4: Montage/Assembly". In: (2020).
- [COLE14] D. Coleman, I. Sucan, S. Chitta, and N. Correll. "Reducing the barrier to entry of complex robotic software: a moveit! case study". In: *arXiv preprint arXiv:1404.3785* (2014).
- [CORV22a] B. Corves, M. Hüsing, S. Bezrucav, M. Lorenz, N. Mandischer, M. Schmitz, A. Shahidi, T. Haschke, C. Weidemann, and T. Huhn. "Robotic 4.0". In: *Handbook Industry 4.0: Law, Technology, Society*. Springer, 2022, pp. 471–490.
- [CORV22b] B. Corves and A. Shahidi. "Kinematic Graph for Motion Planning of Robotic Manipulators". In: *Robotics* 11.5 (2022), p. 105.
- [COX20] J. Cox. *It's never been this hard for companies to find qualified workers*. 2020. URL: <https://www.cnbc.com/2020/02/19/its-never-been-this-hard-for-companies-to-find-qualified-workers.html> (visited on 02/19/2020).
- [COX22] J. Cox. *It's never been this hard for companies to find qualified workers* Diese Grafiken zeigen: So verzweifelt suchen Firmen nach Mitarbeitern. 2022. URL: https://www.wiwo.de/my/erfolg/trends/fuenf-milliarden-fuer-stellenanzeigen-diese-grafiken-zeigen-so-verzweifelt-suchen-firmen-nach-mitarbeitern/28870782.html?utm_source=pocket-newtab-global-de-DE&ticket=ST-1197390-9pBFsuqbXuKTh4feMLYT-cas01.example.org (visited on 02/19/2023).
- [CUTT07] A. F. Cutting-Decelle, R. Young, J. J. Michel, R. Grangel, J. Le Cardinal, and J. P. Bourey. "ISO 15531 MANDATE: A Product-process-resource based Approach for Managing Modularity in Production Management". In: *Concurrent Engineering* 15.2 (2007), pp. 217–235. ISSN: 1063-293X. DOI: 10.1177/1063293X07079329.
- [DAI21] P. Dai, M. Hassan, X. Sun, M. Zhang, Z. Bian, and D. Liu. "A framework for multi-robot coverage analysis of large and complex structures". In: *Journal of Intelligent Manufacturing* (2021). ISSN: 0956-5515. DOI: 10.1007/s10845-021-01745-8.

- [DALI22] M. Dalibor, N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer, and A. Wortmann. "A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins". In: *Journal of Systems and Software* 193 (2022), p. 111361. ISSN: 01641212. DOI: 10.1016/j.jss.2022.111361.
- [DASH05] A. K. Dash, I.-M. Chen, S. H. Yeo, and G. Yang. "Workspace generation and planning singularity-free path for parallel manipulators". In: *Mechanism and Machine Theory* 40.7 (2005), pp. 776–805.
- [DEUT03] Deutsches Institut für Normung e.V., Beuth Verlag GmbH. *DIN 8593, Fertigungsverfahren Fügen, Teil 0: Allgemeines, Einordnung, Unterteilung, Begriffe*. Berlin, 2003.
- [DEUT22] Deutsches Institut für Normung e.V., Beuth Verlag GmbH. *DIN 8580, Fertigungsverfahren - Begriffe, Einteilung*. Berlin, 2022.
- [DEUT95] Deutsches Institut für Normung e.V., Beuth Verlag GmbH. *DIN 1319-1, Grundlagen der Meßtechnik, Teil 1: Grundbegriffe*. Berlin, 1995.
- [DIAB19] M. Diab, A. Akbari, M. Ud Din, and J. Rosell. "PMK-A Knowledge Processing Framework for Autonomous Robotics Perception and Manipulation". In: *Sensors (Basel, Switzerland)* 19.5 (2019). DOI: 10.3390/s19051166.
- [DOAN17] N. C. N. Doan and W. Lin. "Optimal robot placement with consideration of redundancy problem for wrist-partitioned 6R articulated robots". In: *Robotics and Computer-Integrated Manufacturing* 48 (2017), pp. 233–242.
- [DOLG22] A. Dolgui, F. Sgarbossa, and M. Simonetto. "Design and management of assembly systems 4.0: systematic literature review and research agenda". In: *International Journal of Production Research* 60.1 (2022), pp. 184–210. ISSN: 0020-7543. DOI: 10.1080/00207543.2021.1990433.
- [ELAN11] M. Elango, S. Nachiappan, and M. K. Tiwari. "Balancing task allocation in multi-robot systems using K -means clustering and auction based mechanisms". In: *Expert Systems with Applications* 38.6 (2011), pp. 6486–6491. ISSN: 09574174. DOI: 10.1016/j.eswa.2010.11.097.
- [ELMA05] H. A. ElMaraghy. "Flexible and reconfigurable manufacturing systems paradigms". In: *International Journal of Flexible Manufacturing Systems* 17.4 (2005), pp. 261–276. ISSN: 0920-6299. DOI: 10.1007/s10696-006-9028-7.
- [ELMA09] H. A. ElMaraghy and H.-P. Wiendahl. "Changeability—an introduction". In: *Changeable and reconfigurable manufacturing systems* (2009), pp. 3–24.
- [ELMA13] H. ElMaraghy, G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. "Product variety management". In: *Cirp Annals* 62.2 (2013), pp. 629–652.
- [ELMA16] H. ElMaraghy and W. ElMaraghy. "Smart Adaptable Assembly Systems". In: *Procedia CIRP* 44 (2016), pp. 4–13. ISSN: 22128271. DOI: 10.1016/j.procir.2016.04.107.

- [ELMA21] H. ElMaraghy, L. Monostori, G. Schuh, and W. ElMaraghy. "Evolution and future of manufacturing systems". In: *CIRP Annals* 70.2 (2021), pp. 635–658. ISSN: 00078506. DOI: 10.1016/j.cirp.2021.05.008.
- [ESTE96] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.
- [EZUG22] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu. "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), p. 104743. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2022.104743>.
- [FAN21] Q. Fan, Z. Gong, B. Tao, Y. Gao, Z. Yin, and H. Ding. "Base position optimization of mobile manipulators for machining large complex components". In: *Robotics and Computer-Integrated Manufacturing* 70 (2021), p. 102138.
- [FECH18] M. Fechter, C. Seeber, and S. Chen. "Integrated process planning and resource allocation for collaborative robot workplace design". In: *Procedia CIRP* 72 (2018), pp. 39–44.
- [FERR12] P. Ferreira and N. Lohse. "Configuration model for evolvable assembly systems". In: *4th CIRP Conference On Assembly Technologies And Systems*. 2012, p. 3.
- [FORS20] M. Forstenhausler, T. Wetner, and K. Dietmayer. "Optimized Mobile Robot Positioning for better Utilization of the Workspace of an attached Manipulator". In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 2074–2079. ISBN: 978-1-7281-6794-7. DOI: 10.1109/AIM43001.2020.9158922.
- [FREI08] R. Frei, G. D. M. Serugendo, and J. Barata. "Designing self-organization for evolvable assembly systems". In: *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE. 2008, pp. 97–106.
- [FROS22] R. Froschauer, A. Köcher, K. Meixner, S. Schmitt, and F. Spitzer. "Capabilities and skills in manufacturing: a survey over the last decade of ETFA". In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2022, pp. 1–8.
- [GADA17] M. Gadaleta, G. Berselli, and M. Pellicciari. "Energy-optimal layout design of robotic work cells: Potential assessment on an industrial case study". In: *Robotics and Computer-Integrated Manufacturing* 47 (2017), pp. 102–111.
- [GEIH20] K. Geihs. "Engineering challenges ahead for robot teamwork in dynamic environments". In: *Applied Sciences* 10.4 (2020), p. 1368.

- [GINS22] L. V. D. Ginste, A. D. Cock, A. V. Alboom, S. Huysentruyt, E.-H. Aghezzaf, and J. Cottyn. "A formal skill model to enable reconfigurable assembly systems". In: *International Journal of Production Research* 0.0 (2022), pp. 1–16. DOI: 10.1080/00207543.2022.2128922. eprint: <https://doi.org/10.1080/00207543.2022.2128922>.
- [GLIM14] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang. "HermiT: an OWL 2 reasoner". In: *Journal of automated reasoning* 53 (2014), pp. 245–269.
- [GÓME04] A. Gómez-Pérez. "Ontology Evaluation". In: *Handbook on Ontologies*. Ed. by S. Staab and R. Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 251–273. ISBN: 978-3-662-11957-0. DOI: 10.1007/978-3-540-24750-0{\textunderscore}13.
- [GÖPP21] A. Göppert, L. Grahn, J. Rachner, D. Grunert, S. Hort, and R. H. Schmitt. "Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems". In: *Journal of Intelligent Manufacturing* (2021). ISSN: 0956-5515. DOI: 10.1007/s10845-021-01860-6.
- [GÖPP24] A. Göppert, C. Schou, A. Kluge-Wilkes, L. Bergs, I. Rodriguez, L. Schäper, H. Arnason, D. Arias-Cachero Rincon, N. König, O. Madsen, P. Mogensen, and R. H. Schmitt. *5G Edge-Cloud-based Robotics for Flexible and Resilient Production Systems*. (In review). 2024.
- [GOSS91] C. Gosselin and J. Angeles. "A global performance index for the kinematic optimization of robotic manipulators". In: (1991).
- [GRUB93] T. R. Gruber. "A translation approach to portable ontology specifications". In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220. ISSN: 10428143. DOI: 10.1006/knac.1993.1008.
- [GUAR09] N. Guarino, D. Oberle, and S. Staab. "What is an ontology?" In: *Handbook on ontologies*. Springer, 2009, pp. 1–17.
- [GUPT82] K. Gupta and B. Roth. "Design considerations for manipulator workspace". In: (1982).
- [HASC23] T. Haschke. "Planning of indoor construction tasks for mobile manipulators". Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2023. Dissertation. Aachen: Rheinisch-Westfälische Technische Hochschule Aachen, 2023, 1 Online-Ressource : Illustrationen, Diagramme. DOI: 10.18154/RWTH-2023-08445.
- [HASS15] M. Hassan, D. Liu, G. Paul, and S. Huang. "An approach to base placement for effective collaboration of multiple autonomous industrial robots". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3286–3291. ISBN: 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139652.
- [HASS16] M. Hassan, D. Liu, and G. Paul. "Modeling and stochastic optimization of complete coverage under uncertainties in multi-robot base placements". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2978–2984.

- [HASS18] M. Hassan, D. Liu, and G. Paul. "Collaboration of Multiple Autonomous Industrial Robots through Optimal Base Placements". In: *Journal of Intelligent & Robotic Systems* 90.1-2 (2018), pp. 113–132. ISSN: 0921-0296. DOI: 10.1007/s10846-017-0647-x.
- [HATT22] R. van Hattem. *numpy-stl*. 2022.
- [HATV78] J. Hatvany and L. Nemes. "Intelligent manufacturing systems—a tentative forecast". In: *IFAC Proceedings Volumes* 11.1 (1978), pp. 895–899.
- [HENR22] P. Henrich. *Daten und Fakten zum Fachkräftemangel in Deutschland*. 2022. URL: https://de.statista.com/themen/887/fachkraeftemangel/#dossierContents__outerWrapper (visited on 11/28/2022).
- [HEVN03] A. R. Hevner and S. T. March. "The information systems research cycle". In: *Computer* 36.11 (2003), pp. 111–113.
- [HEVN04] Hevner, March, Park, and Ram. "Design Science in Information Systems Research". In: *MIS Quarterly* 28.1 (2004), p. 75. ISSN: 02767783. DOI: 10.2307/25148625.
- [HEVN07] A. R. Hevner. "A Three Cycle View of Design Science Research". In: *Scandinavian Journal of Information Systems* Vol. 19 : Iss. 2 (2007), Article 4.
- [HOFF17] M. Hoffmann. *Adaptive und skalierbare Informationsmodellierung zur Ermöglichung autonomer Entscheidungsfindungsprozesse für interoperable, reaktive Fertigungen*. 2017. DOI: 10.18154/RWTH-2017-07373.
- [HU11] S. J. Hu, J. Ko, L. Weyand, H. A. ElMaraghy, T. K. Lien, Y. Koren, H. Bley, G. Chryssolouris, N. Nasr, and M. Shpitalni. "Assembly system design and operations for product variety". In: *CIRP Annals* 60.2 (2011), pp. 715–733. ISSN: 00078506. DOI: 10.1016/j.cirp.2011.05.004.
- [HU13] S. J. Hu. "Evolving Paradigms of Manufacturing: From Mass Production to Mass Customization and Personalization". In: *Procedia CIRP* 7 (2013), pp. 3–8. ISSN: 22128271. DOI: 10.1016/j.procir.2013.05.002.
- [HUAN21] Y. Huang, S. Dhouib, and J. Malenfant. "An AAS Modeling Tool for Capability-Based Engineering of Flexible Production Lines". In: *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2021, pp. 1–6.
- [HUO08] L. Huo and L. Baron. "The joint-limits and singularity avoidance in robotic welding". In: *Industrial Robot: An International Journal* 35.5 (2008), pp. 456–464.
- [HÜTT19] G. Hüttemann, A. F. Buckhorst, and R. H. Schmitt. "Modelling and Assessing Line-less Mobile Assembly Systems". In: *Procedia CIRP* 81 (2019), pp. 724–729. ISSN: 22128271. DOI: 10.1016/j.procir.2019.03.184.

- [HÜTT21] G. Hüttemann. "Model-based a priori analysis of line-less mobile assembly systems; 1". Druckausgabe: 2021. - Auch veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2020. Dissertation. Aachen: RWTH Aachen University, 2021, 1 Online-Ressource (xix, 293 Seiten) : Illustrationen, Diagramme. ISBN: 978-3-86359-943-0. DOI: 10.18154/RWTH-2021-02011.
- [IIVA07] J. Iivari. "A Paradigmatic Analysis of Information Systems As a Design Science". In: *Scand. J. Inf. Syst.* 19 (2007), p. 5.
- [JANA17] F. Janati, F. Abdollahi, S. S. Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad. "Multi-robot Task Allocation Using Clustering Method". In: *Robot Intelligence Technology and Applications 4*. Ed. by J.-H. Kim, F. Karray, J. Jo, P. Sincak, and H. Myung. Vol. 447. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2017, pp. 233–247. ISBN: 978-3-319-31291-0. DOI: 10.1007/978-3-319-31293-4{\textunderscore}19.
- [JÄRV19] E. Järvenpää, N. Siltala, O. Hylli, and M. Lanz. "The development of an ontology for describing the capabilities of manufacturing resources". In: *Journal of Intelligent Manufacturing* 30.2 (2019), pp. 959–978. ISSN: 0956-5515. DOI: 10.1007/s10845-018-1427-6.
- [KALA18] R. Kalawoun, S. Lengagne, and Y. Mezouar. "Optimal robot base placements for coverage tasks". In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2018, pp. 235–240.
- [KALA19] R. Kalawoun. "Motion planning of multi-robot system for airplane stripping". PhD thesis. Université Clermont Auvergne [2017-2020], 2019.
- [KATA19] B. Katalinic, D. Haskovic, I. Kukushkin, and I. Zec. "Hybrid control structure and reconfiguration capabilities in bionic assembly system". In: *Advanced Control Techniques in Complex Engineering Systems: Theory and Applications: Dedicated to Professor Vsevolod M. Kuntsevich* (2019), pp. 259–277.
- [KAWA22] T. Kawabe, Z. Liu, T. Nishi, M. M. Alam, and T. Fujiwara. "Optimal Motion Planning and Layout Design in Robotic Cellular Manufacturing Systems". In: *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE. 2022, pp. 1541–1545.
- [KIM91] J.-O. Kim and K. Khosla. "Dexterity measures for design and control of manipulators". In: *Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91*. IEEE. 1991, pp. 758–763.
- [KLEI87] C. A. Klein and B. E. Blaho. "Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators". In: *The International Journal of Robotics Research* 6.2 (1987), pp. 72–83. ISSN: 0278-3649. DOI: 10.1177/027836498700600206.

- [KLUG11] S. Kluge and E. Westkämper. *Methodik zur fähigkeitsbasierten Planung modularer Montagesysteme: Zugl.: Stuttgart, Universität Stuttgart, Diss., 2011*. Vol. 510. IPA-IAO-Forschung und Praxis. Stuttgart: Universitätsbibliothek der Universität Stuttgart, 2011. ISBN: 978-3-939890-81-2.
- [KLUG22a] A. Kluge-Wilkes and B. Gunaseelan. *Ontology-based task allocation for heterogeneous resources in Line-less Mobile Assembly Systems*. https://git-ce.rwth-aachen.de/wzl-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/capilano. 2022.
- [KLUG22b] A. Kluge-Wilkes, B. Gunaseelan, and R. H. Schmitt. "Ontology-based task allocation for heterogeneous resources in line-less mobile assembly systems". In: *MHI Colloquium*. Springer. 2022, pp. 53–63. DOI: https://doi.org/10.1007/978-3-031-10071-0_5.
- [KLUG22c] A. Kluge-Wilkes and R. H. Schmitt. "Dynamic Task Allocation for Cooperating, Heterogeneous Assembly Resources in LMAS". In: *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems*. Ed. by A.-L. Andersen, R. Andersen, T. D. Brunoe, M. S. S. Larsen, K. Nielsen, A. Napoleone, and S. Kjeldgaard. Springer eBook Collection. Cham: Springer International Publishing and Imprint Springer, 2022, pp. 535–542. ISBN: 978-3-030-90700-6. DOI: https://doi.org/10.1007/978-3-030-90700-6_60.
- [KLUG22d] A. Kluge-Wilkes and R. H. Schmitt. "Mobile Robot Base Placement for Assembly Systems: Survey, Measures and Task Clustering". In: *Production at the Leading Edge of Technology*. Ed. by B.-A. Behrens, A. Brosius, W.-G. Drossel, W. Hintze, S. Ihlenfeldt, and P. Nyhuis. Lecture Notes in Production Engineering. Cham: Springer International Publishing, 2022, pp. 439–447. ISBN: 978-3-030-78423-2. DOI: [10.1007/978-3-030-78424-9_49](https://doi.org/10.1007/978-3-030-78424-9_49).
- [KLUG23a] A. Kluge-Wilkes. *Autonomous Clustering-based Task Allocation and Base Placement for Mobile Robots in Line-less Mobile Assembly Systems*. 2023. DOI: [10.36227/techrxiv.19773352.v12312321321323123](https://doi.org/10.36227/techrxiv.19773352.v12312321321323123).
- [KLUG23b] A. Kluge-Wilkes. *End-effector dependent executability*. https://git-ce.rwth-aachen.de/wzl-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/end-effector-dependent-executability. 2023.
- [KLUG23c] A. Kluge-Wilkes. *Extending the executability of assembly task poses by robot through end-effectors*. Version 1.0. Zenodo, Dec. 2023. DOI: [10.5281/zenodo.10119467](https://doi.org/10.5281/zenodo.10119467).
- [KLUG23d] A. Kluge-Wilkes, R. Baier, I. Kunze, A. Müller, A. Shahidi, D. Wolfschläger, C. Brecher, B. Corves, M. Hüsing, V. Nitsch, R. H. Schmitt, and K. Wehrl. "Modular Control and Services to Operate Lineless Mobile Assembly Systems". In: *Internet of Production: Fundamentals, Applications and Proceedings*. Ed. by C. Brecher, G. Schuh, W. van der Aalst, M. Jarke, F. T. Piller, and M. Padberg. Cham: Springer International Publishing, 2023, pp. 1–26. ISBN: 978-3-030-98062-7. DOI: [10.1007/978-3-030-98062-7_13-2](https://doi.org/10.1007/978-3-030-98062-7_13-2).

- [KLUG23e] A. Kluge-Wilkes, A. Göppert, and R. H. Schmitt. "End-Effector-Dependent Evaluation of Task Executability by Mobile Robots in Changeable Assembly Stations". In: *Proceedings of the Changeable, Agile, Reconfigurable and Virtual Production Conference and the World Mass Customization & Personalization Conference*. Springer. 2023, pp. 536–543. DOI: https://doi.org/10.1007/978-3-031-34821-1_58.
- [KLUG24a] A. Kluge-Wilkes. *Evaluating robot workspaces with performance maps*. Zenodo, Dec. 2024. DOI: <https://zenodo.org/records/10462326>.
- [KLUG24b] A. Kluge-Wilkes, P. D. Reverdito, S. Kohar, A. Göppert, and R. H. Schmitt. "Evaluating task executability of mobile robots with performance maps". In: *Procedia CIRP* 127 (2024). 10th CIRP Conference on Assembly Technology and Systems (CIRP CATS 2024), pp. 38–43. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2024.07.008>.
- [KORE10] Y. Koren. *The global manufacturing revolution: product-process-business integration and reconfigurable systems*. John Wiley & Sons, 2010.
- [KORE99] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. "Reconfigurable manufacturing systems". In: *CIRP annals* 48.2 (1999), pp. 527–540.
- [KORN22] T. Korn and H. Stemmler. "Russia's war against Ukraine might persistently shift global supply chains". In: *VoxEU.org* 31 (2022).
- [KÖVE07] J. Kövecses and J. Angeles. "The stiffness matrix in elastically articulated rigid-body systems". In: *Multibody System Dynamics* 18 (2007), pp. 169–184.
- [KUBI77] H. Kubicek. "Empirische und handlungstheoretische Forschungskonzeption in der Betriebswirtschaftslehre". In: *Empirische und handlungstheoretische Forschungskonzeption in der Betriebswirtschaftslehre*. Stuttgart: Poeschel (1977), pp. 3–36.
- [KUMA17] G. A. Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai. "A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification". In: *Sensors (Basel, Switzerland)* 17.6 (2017). DOI: 10.3390/s17061268.
- [KUMA81] A. Kumar and K. J. Waldron. "The Workspaces of a Mechanical Manipulator". In: *Journal of Mechanical Design* 103.3 (1981), pp. 665–672. ISSN: 0161-8458. DOI: 10.1115/1.3254968.
- [KUNZ11] L. Kunze, T. Roehm, and M. Beetz. "Towards semantic robot description languages". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. IEEE, 2011, pp. 5589–5595. ISBN: 1612843859.
- [KUNZ17] M. Kunze. *On-the-Fly Workspace Visualization for Redundant Manipulators*. Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2016, 2017.
- [LAVA06] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.

- [LEE89] S. Lee. "Dual redundant arm configuration optimization with task-oriented dual arm manipulability". In: *IEEE Transactions on Robotics and Automation* 5.1 (1989), pp. 78–97.
- [LEIT06] P. Leitão and F. Restivo. "ADACOR: A holonic architecture for agile and adaptive manufacturing control". In: *Computers in industry* 57.2 (2006), pp. 121–130.
- [LEIT08] P. Leitão and F. J. Restivo. "Implementation of a holonic control system in a flexible manufacturing system". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.5 (2008), pp. 699–709.
- [LEMA06] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko. "MASON: A Proposal For An Ontology Of Manufacturing Domain". In: *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*. IEEE, 2006, pp. 195–200. ISBN: 0-7695-2589-X. DOI: 10.1109/DIS.2006.48.
- [LIAO17] Y. Liao, F. Deschamps, E. d. F. R. Loures, and L. F. P. Ramos. "Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal". In: *International journal of production research* 55.12 (2017), pp. 3609–3629.
- [LIEB21] M. R. Liebenberg, G. Hirt, M. Jarke, and G. Lakemeyer. *Autonomous agents for the World Wide Lab: artificial intelligence in the manufacturing industry*. Tech. rep. Fachgruppe Informatik, 2021.
- [LIM17] Z. Y. Lim, S. Ponnambalam, and K. Izui. "Multi-objective hybrid algorithms for layout optimization in multi-robot cellular manufacturing systems". In: *Knowledge-Based Systems* 120 (2017), pp. 87–98.
- [LIND05] S. R. Lindemann and S. M. LaValle. "Current issues in sampling-based motion planning". In: *Robotics research. The eleventh international symposium*. Springer, 2005, pp. 36–54.
- [LOTT06] B. Lotter and H.-P. Wiendahl, eds. *Montage in der industriellen Produktion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-21413-7. DOI: 10.1007/3-540-36669-5.
- [LOTT12] B. Lotter and H.-P. Wiendahl. *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. 2. Aufl. 2013. VDI-Buch. Berlin and Heidelberg: Springer, 2012. ISBN: 978-3-642-29061-9. DOI: 10.1007/978-3-642-29061-9.
- [LOVR08] S. Lovrencic and M. Cubrilo. "Ontology Evaluation - Comprising Verification and Validation". In: *Central European Conference on Information and Intelligent Systems*; ed. by Faculty of Organization and Informatics Varazdin. Croatia, Varazdin, 2008.
- [LUET92] T. C. Lueth. "Automated planning of robot workcell layouts". In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE, 1992, pp. 1103–1108.
- [MABK20] M. M. Mabkhot, S. K. Amri, S. Darmoul, A. M. Al-Samhan, and S. Elkosantini. "An ontology-based multi-criteria decision support system to reconfigure manufacturing systems". In: *IJSE Transactions* 52.1 (2020), pp. 18–42.

- [MAKH18] A. Makhal and A. K. Goins. "Reuleaux: Robot Base Placement by Reachability Analysis". In: *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 137–142. ISBN: 978-1-5386-4652-6. DOI: 10.1109/IRC.2018.00028.
- [MAKR12] S. Makris, G. Michalos, A. Eytan, and G. Chryssolouris. "Cooperating Robots for Reconfigurable Assembly Operations: Review and Challenges". In: *Procedia CIRP* 3 (2012), pp. 346–351. ISSN: 22128271. DOI: 10.1016/j.procir.2012.07.060.
- [MALA18] S. Malakuti, J. Bock, M. Weser, P. Venet, P. Zimmermann, M. Wiegand, J. Grothoff, C. Wagner, and A. Bayha. "Challenges in skill-based engineering of industrial automation systems". In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. IEEE. 2018, pp. 67–74.
- [MALH19a] R. Malhan, a. Kabir, B. Shah, and S. Gupta. "Identifying Feasible Workpiece Placement with Respect to Redundant Manipulator for Complex Manufacturing Tasks". In: *2019 International Conference on Robotics and Automation (ICRA)*. [Piscataway, NJ]: IEEE, 2019. ISBN: 978-1-5386-6027-0.
- [MALH19b] R. K. Malhan, A. M. Kabir, B. Shah, T. Centea, and S. K. Gupta. "Determining feasible robot placements in robotic cells for composite prepreg sheet layup". In: *International Manufacturing Science and Engineering Conference*. Vol. 58745. American Society of Mechanical Engineers. 2019, V001T02A025.
- [MALH21] R. K. Malhan, A. V. Shembekar, A. M. Kabir, P. M. Bhatt, B. Shah, S. Zanio, S. Nutt, and S. K. Gupta. "Automated planning for robotic layup of composite prepreg". In: *Robotics and computer-integrated manufacturing* 67 (2021), p. 102020.
- [MALH22] R. Malhan and S. K. Gupta. "Finding Optimal Sequence of Mobile Manipulator Placements for Automated Coverage Planning of Large Complex Parts". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 86212. American Society of Mechanical Engineers. 2022, V002T02A006.
- [MANS22] N. Mansfeld, G. G. Peña, M. Hamad, A. Kurdas, S. Abdolshah, and S. Haddadin. "Global Safety Characteristics of Wheeled Mobile Manipulators". In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. 2022, pp. 2245–2252. DOI: 10.1109/CASE49997.2022.9926607.
- [MARC95] S. T. March and G. F. Smith. "Design and natural science research on information technology". In: *Decision support systems* 15.4 (1995), pp. 251–266.
- [MART03] P. Martin and A. D'Acunto. "Design of a production system: An application of integration product-process". In: *International Journal of Computer Integrated Manufacturing* 16.7-8 (2003), pp. 509–516. ISSN: 0951-192X. DOI: 10.1080/0951192031000115831.

- [MITS08] S. Mitsi, K.-D. Bouzakis, D. Sagris, and G. Mansour. "Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm". In: *Robotics and Computer-Integrated Manufacturing* 24.1 (2008), pp. 50–59. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2006.08.003>.
- [MOHE10] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, P. Group, et al. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement". In: *International journal of surgery* 8.5 (2010), pp. 336–341.
- [MONO15] L. Monostori. In: *at - Automatisierungstechnik* 63.10 (2015), pp. 766–776. DOI: [doi:10.1515/auto-2015-0066](https://doi.org/10.1515/auto-2015-0066).
- [MÜLL15] R. Müller, J. Eilers, L. Hörauf, and L. Hermanns. "Modell zur fähigkeitsbasierten Montageplanung". In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 110.9 (2015), pp. 553–557. ISSN: 0947-0085. DOI: [10.3139/104.111393](https://doi.org/10.3139/104.111393).
- [MÜNK21] S. Münker, P. R. Wildemann, A. Göppert, S. Brell-Cokcan, and R. H. Schmitt. "Online capability-based resource allocation for on-site construction operations utilizing digital twin models". In: *Construction Robotics* 5.3-4 (2021), pp. 211–226.
- [MUTT21] S. Mutti, G. Nicola, M. Beschi, N. Pedrocchi, and L. M. Tosatti. "Towards optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms". In: *Robotics and Computer-Integrated Manufacturing* 71 (2021), p. 102131. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2021.102131>.
- [NATI75] National Bureau of Standards, ed. *Performance Evaluation of Programmable Robots and Manipulators: Report of a Workshop*. Annapolis, Maryland, 1975.
- [NBUX69] J. N. Buxton and B. Randell. *SOFTWARE ENGINEERING TECHNIQUES: Report on a conference sponsored by the NATO SCIENCE COMMITTEE*. Ed. by NATO Science Committee. England at The Kynoch Press, Birmingham, Oct. 1, 1969.
- [NEKT10] A. Nektarios and N. A. Aspragathos. "Optimal location of a general position and orientation end-effector's path relative to manipulator's base, considering velocity performance". In: *Robotics and Computer-Integrated Manufacturing* 26.2 (2010), pp. 162–173.
- [NICK20] C. Nicksch, A. Kluge-Wilkes, M. Huber, and R. H. Schmitt. "Global Reference System for factory-wide integration of metrology enabling flexible automation in aeroplane assembly—requirements, concept and suitable technologies". In: *Procedia Manufacturing* 52 (2020), pp. 89–94. DOI: <https://doi.org/10.1016/j.promfg.2020.11.017>,.
- [NILE01] I. Niles and A. Pease. "Towards a standard upper ontology". In: *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*. 2001, pp. 2–9.

- [NORM16] D. D. I. für Normung e. V. *DIN SPEC 91345:2016-04, Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*. Berlin, 2016. DOI: <https://dx.doi.org/10.31030/2436156>.
- [NOY01] N. Noy and D. L. McGuinness. "Ontology development 101". In: *Knowledge Systems Laboratory, Stanford University 2001* (2001).
- [OLIV19] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. Goncalves, M. K. Habib, J. Bermejo-Alonso, M. Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton, A. Gyrard, S. Borgo, G. Alenyà, M. Beetz, and H. Li. "A review and comparison of ontology-based approaches to robot autonomy". In: *The Knowledge Engineering Review* 34 (2019). ISSN: 0269-8889. DOI: 10.1017/S0269888919000237.
- [ONOR06] M. Onori, J. Barata, and R. Frei. "Evolvable assembly systems basic principles". In: *International Conference on Information Technology for Balanced Automation Systems*. Springer. 2006, pp. 317–328.
- [ONOR10] M. Onori and J. Barata. "Evolvable production systems: Mechatronic production equipment with evolutionary control". In: *IFIP Advances in Information and Communication Technology, 2010* (2010).
- [OSMA97] Osman Balci. "VERIFICATION, VALIDATION AND ACCREDITATION OF SIMULATION MODELS". In: *Proceedings of the 1997 Winter Simulation Conference*. Ed. by S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson. 1997, pp. 135–141.
- [PAND17] A. Pandey, S. Pandey, and D. Parhi. "Mobile robot navigation and obstacle avoidance techniques: A review". In: *Int Rob Auto J* 2.3 (2017), p. 00022.
- [PATE15] S. Patel and T. Sobh. "Manipulator Performance Measures - A Comprehensive Literature Survey". In: *Journal of Intelligent & Robotic Systems* 77.3-4 (2015), pp. 547–570. ISSN: 0921-0296. DOI: 10.1007/s10846-014-0024-y.
- [PAUS17] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour. "A combined approach for robot placement and coverage path planning for mobile manipulation". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 6285–6292.
- [PEDE16] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen. "Robot skills for manufacturing: From concept to industrial deployment". In: *Robotics and Computer-Integrated Manufacturing* 37 (2016), pp. 282–291.
- [PEDR11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.
- [PELL15] S. Pellegrinelli, S. Borgia, N. Pedrocchi, E. Villagrossi, G. Bianchi, and L. M. Tosatti. "Minimization of the energy consumption in motion planning for single-robot tasks". In: *Procedia Cirp* 29 (2015), pp. 354–359.

- [PELL17] S. Pellegrinelli, N. Pedrocchi, L. M. Tosatti, A. Fischer, and T. Tolio. "Multi-robot spot-welding cells for car-body assembly: Design and motion planning". In: *Robotics and Computer-Integrated Manufacturing* 44 (2017), pp. 97–116.
- [PENN19a] J. Pennekamp, R. Glebke, M. Henze, T. Meisen, C. Quix, R. Hai, L. Gleim, P. Niemietz, M. Rudack, S. Knappe, A. Epple, D. Trauth, U. Vroomen, T. Bergs, C. Brecher, A. Buhrig-Polaczek, M. Jarke, and K. Wehrle. "Towards an Infrastructure Enabling the Internet of Production". In: *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 31–37. ISBN: 978-1-5386-8500-6. DOI: 10.1109/ICPHYS.2019.8780276.
- [PENN19b] J. Pennekamp, R. Glebke, M. Henze, T. Meisen, C. Quix, R. Hai, L. Gleim, P. Niemietz, M. Rudack, S. Knappe, et al. "Towards an infrastructure enabling the internet of production". In: *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE. 2019, pp. 31–37.
- [PERZ19] A. Perzylo, J. Grothoff, L. Lucio, M. Weser, S. Malakuti, P. Venet, V. Aravantinos, and T. Deppe. "Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective". In: *IFAC-PapersOnLine* 52.13 (2019), pp. 1590–1596. ISSN: 24058963. DOI: 10.1016/j.ifacol.2019.11.427.
- [PETE08] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. "Systematic mapping studies in software engineering". In: *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)* 12. 2008, pp. 1–10.
- [PFRO13] J. Pfrommer, M. Schleipen, and J. Beyerer. "PPRS: Production skills and their relation to product, process, and resource". In: *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2013, pp. 1–4.
- [PUGL12] L. J. Puglisi, R. J. Saltaren, H. A. Moreno, P. F. Cárdenas, C. Garcia, and R. Aracil. "Dimensional synthesis of a spherical parallel manipulator based on the evaluation of global performance indexes". In: *Robotics and Autonomous Systems* 60.8 (2012), pp. 1037–1045.
- [PUGL19] R. Pugliese, T. Konrad, R. Zweigel, A. Kluge-Wilkes, and R. Schmitt. "Automatische Multikopter-basierte Indoor-Inspektion großer Oberflächen". In: *Automation 2019*. Ed. by V. W. GmbH. Vol. 2351. VDI-Berichte. Düsseldorf: VDI Verlag, 2019. DOI: <https://doi.org/10.51202/9783181023518-767>.
- [RABE08] M. Rabe, S. Spieckermann, and S. Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. 1. Aufl. Berlin and Heidelberg: Springer, 2008. ISBN: 978-3-540-35282-2.
- [RESE] ResearchGate. *Website ResearchGate*. URL: www.researchgate.net (visited on 06/06/2023).
- [RIZK19] Y. Rizk, M. Awad, and E. W. Tunstel. "Cooperative heterogeneous multi-robot systems: A survey". In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–31.
- [ROBO15] I. Robotics and A. Society. "IEEE standard ontologies for robotics and automation". In: *IEEE Stan.* 1872 (2015), pp. 1–60.

- [RODR21] I. Rodriguez, R. S. Mogensen, A. Schjørring, M. Razzaghpour, R. Maldonado, G. Berardinelli, R. Adeogun, P. H. Christensen, P. Mogensen, O. Madsen, et al. "5G swarm production: Advanced industrial manufacturing concepts enabled by wireless automation". In: *IEEE Communications Magazine* 59.1 (2021), pp. 48–54.
- [ROLD21] J. J. Roldán-Gómez and A. Barrientos. *Special issue on multi-robot systems: Challenges, trends, and applications*. 2021.
- [ROLD22] J. J. Roldán Gómez and A. Barrientos. "Multi-Robot Systems: Challenges, Trends and Applications". In: (2022).
- [ROTH75] Roth, Bernard. "Performance evaluation of mechanical manipulators". In: *Performance Evaluation of Programmable Robots and Manipulators*. Ed. by National Bureau of Standards. Annapolis, Maryland, 1975, pp. 39–62.
- [ROZO17] L. Rozo, N. Jaquier, S. Calinon, and D. G. Caldwell. "Learning manipulability ellipsoids for task compatibility in robot manipulation". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 3183–3189.
- [RUSS22] M. Russo. "Measuring Performance: Metrics for Manipulator Design, Control, and Optimization". In: *Robotics* 12.1 (2022), p. 4.
- [SALI82] J. K. Salisbury and J. J. Craig. "Articulated hands: Force control and kinematic issues". In: *The International journal of Robotics research* 1.1 (1982), pp. 4–17.
- [SARG10] R. G. Sargent. "Proceedings of the 2010 Winter Simulation Conference (WSC 2010): Baltimore, Maryland, USA, 5 - 8 December 2010 ; [incorporating the MASM (Modeling and Analysis for Semiconductor Manufacturing) Conference". In: *Proceedings of the 2010 Winter Simulation Conference*. Piscataway, NJ: IEEE, 2010, pp. 166–183. ISBN: 978-1-4244-9866-6. DOI: 10.1109/WSC.2010.5679166.
- [SARG96] R. G. Sargent. "Some subjective validation methods using graphical displays of data". In: *Proceedings of the 28th conference on Winter simulation*. 1996, pp. 345–351.
- [SCHL79] S. Schlesinger, R. Crosbier, R. Gagne, G. Innis, Lalwani, C.S., J. Loch, R. Sylvester, R. Wright, N. Kheir, and D. Bartos. "Terminology for model credibility: Reports of SCS Technical Committees". In: *SIMULATION* 32.3 (1979), pp. 103–104. ISSN: 0037-5497. DOI: 10.1177/003754977903200304.
- [SCHO07] B. Scholz-Reiter and M. Freitag. "Autonomous Processes in Assembly Systems". In: *CIRP Annals* 56.2 (2007), pp. 712–729. ISSN: 00078506. DOI: 10.1016/j.cirp.2007.10.002.
- [SCHO18] C. Schou, R. S. Andersen, D. Chrysostomou, S. Bøgh, and O. Madsen. "Skill-based instruction of collaborative robots in industrial settings". In: *Robotics and Computer-Integrated Manufacturing* 53 (2018), pp. 72–80.

- [SCHO22] C. Schou, A. Avhad, S. Bøgh, and O. Madsen. "Towards the swarm production paradigm". In: *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems: Proceedings of the 8th Changeable, Agile, Reconfigurable and Virtual Production Conference (CARV2021) and the 10th World Mass Customization & Personalization Conference (MCPC2021), Aalborg, Denmark, October/November 2021*. Springer. 2022, pp. 105–112.
- [SCHU18] E. Schubert and P. J. Rousseeuw. "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms". In: *CoRR* abs/1810.05691 (2018). arXiv: 1810.05691.
- [SCOP] Scopus. *Website Scopus*. URL: www.scopus.com (visited on 06/06/2023).
- [SETH90] A. K. Sethi and S. P. Sethi. "Flexibility in Manufacturing: A Survey". In: *The International Journal of Flexible Manufacturing Systems*. 2 2 (1990), pp. 289–328.
- [SEUR12] S. Seuring and S. Gold. "Conducting content-analysis based literature reviews in supply chain management". In: *Supply Chain Management: An International Journal* 17.5 (2012), pp. 544–555.
- [SHAH22] A. Shahidi, T. Kinzig, M. Hüsing, and B. Corves. "Kinematically Adapted Sampling-Based Motion Planning Algorithm for Robotic Manipulators". In: *International Symposium on Advances in Robot Kinematics*. Springer. 2022, pp. 453–461.
- [SHAR22a] A. Sharma. *100,000 Mobile Robots Shipped In 2021*. 2022. URL: <https://interactanalysis.com/insight/100000-mobile-robots-shipped-in-2021/> (visited on 11/01/2023).
- [SHAR22b] A. Sharma. *Mobile robot shipments grow by 53 % in 2022*. 2022. URL: <https://www.interactanalysis.com/mobile-robot-shipments-grow-by-53-in-2022/> (visited on 12/2022).
- [SHI19] R. Shi, P. Steenkiste, and M. M. Veloso. "SC-M*: A multi-agent path planning algorithm with soft-collision constraint on allocation of common resources". In: *Applied Sciences* 9.19 (2019), p. 4037.
- [SICI16] B. Siciliano and O. Khatib. *Springer handbook of robotics*. 2nd edition. Berlin and Heidelberg: Springer, 2016. ISBN: 9783319325521. DOI: 10.1007/978-3-319-32550-7.
- [SIMO96] H. A. Simon. *The Sciences of the Artificial*. 3rd. 1996.
- [SISO12] D. Sisodia, L. Singh, S. Sisodia, and K. Saxena. "Clustering techniques: a brief survey of different clustering algorithms". In: *International Journal of Latest Trends in Engineering and Technology (IJLTET)* 1.3 (2012), pp. 82–87.
- [SOET20] P. Soetens, T. Issaris, H. Bruyninckx, S. Joyeux, and R. Smits. *Kinematics Dynamics Library (KDL). Orocos Kinematics and Dynamics*. Accessed = 2023-09-20. 2020.

- [SON19] S.-W. Son and D.-S. Kwon. "A convex programming approach to the base placement of a 6-DOF articulated robot with a spherical wrist". In: *The International Journal of Advanced Manufacturing Technology* 102.9-12 (2019), pp. 3135–3150. ISSN: 0268-3768. DOI: 10.1007/s00170-019-03391-0.
- [SPAR72] K. Sparck Jones. "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of documentation* 28.1 (1972), pp. 11–21.
- [SPEN16] D. Spensieri, J. S. Carlson, R. Bohlin, J. Kressin, and J. Shi. "Optimal robot placement for tasks execution". In: *Procedia CIRP* 44 (2016), pp. 395–400.
- [SPIC02] P. Spicer, Y. Koren, M. Shpitalni, and D. Yip-Hoi. "Design principles for machining system configurations". In: *CIRP Annals* 51.1 (2002), pp. 275–280.
- [STAN21] I. O. for Standardization. *Robotics — Vocabulary*. ISO 8373:2021(E). Vernier, Geneva, Switzerland: International Organization for Standardization, 2021.
- [STAT22] Statista. *Daten und Fakten zum Fachkräftemangel in Deutschland*. 2022. URL: <https://de.statista.com/statistik/daten/studie/1324209/umfrage/fachkraeftemangel-im-verarbeitenden-gewerbe/> (visited on 08/04/2022).
- [STEI97] P. Steinwasser. *Modulares Informationsmanagement in der integrierten Produkt- und Prozessplanung*. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 1997.
- [STOC98] L. Stocco, S. E. Salcudean, and F. Sassani. "Fast constrained global minimax optimization of robot parameters". In: *Robotica* 16.6 (1998), pp. 595–605.
- [STUD98] R. Studer, V. Benjamins, and D. Fensel. "Knowledge engineering: Principles and methods". In: *Data & Knowledge Engineering* 25.1-2 (1998), pp. 161–197. ISSN: 0169023X. DOI: 10.1016/S0169-023X(97)00056-6.
- [STUR90] R. Sturges. "A quantification of machine dexterity applied to an assembly task". In: *The International Journal of Robotics Research* 9.3 (1990), pp. 49–62.
- [SUEM16] I. Suemitsu, K. Izui, T. Yamada, S. Nishiwaki, A. Noda, and T. Nagatani. "Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems". In: *Computers & Industrial Engineering* 102 (2016), pp. 396–407.
- [SURE09] Y. Sure, S. Staab, and R. Studer. "Ontology Engineering Methodology". In: *Handbook on Ontologies*. Ed. by S. Staab and R. Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 135–152. ISBN: 978-3-540-70999-2. DOI: 10.1007/978-3-540-92673-3{\\textunderscore}6.
- [SWIN06] R. Swinbank and R. James Purser. "Fibonacci grids: A novel approach to global modelling". In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 132.619 (2006), pp. 1769–1793.
- [TANE00] T. Tanev and B. Stoyanov. "On the performance indexes for robot manipulators". In: *Problems of engineering cybernetics and robotics* 49 (2000), pp. 64–71.

- [TANE98] T. Tanev and I. Chavdarov. "Performance evaluation of manipulation system and graphical representation of the characteristics". In: *Journal of Theoretical and Applied Mechanics, Sofia* (1998).
- [TANT17] E. Tantik and R. Anderl. "Integrated data model and structure for the asset administration shell in industrie 4.0". In: *Procedia Cirp* 60 (2017), pp. 86–91.
- [TEN09] M. Tenorth and M. Beetz. "KnowRob—knowledge processing for autonomous personal robots". In: *2009 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2009, pp. 4261–4266.
- [TEN13] M. Tenorth and M. Beetz. "KnowRob: A knowledge processing infrastructure for cognition-enabled robots". In: *The International Journal of Robotics Research* 32.5 (2013), pp. 566–590.
- [TERK09] W. Terkaj, T. Tolio, and A. Valente. "A review on manufacturing flexibility". In: *Design of flexible production systems: Methodologies and tools* (2009), pp. 41–61.
- [TOLA22] D. Tola, E. Madsen, C. Gomes, L. Esterle, C. Schlette, C. Hansen, and P. G. Larsen. "Towards Easy Robot System Integration: Challenges and Future Directions". In: *2022 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2022, pp. 77–82. ISBN: 978-1-6654-4540-5. DOI: 10.1109/SII52469.2022.9708846.
- [TRAB22] A. Trabelsi, J. Sandoval, A. Mlika, S. Lahouar, S. Zeghloul, J. Cau, and M. A. Laribi. "Optimal Multi-robot Placement Based on Capability Map for Medical Applications". In: *Advances in Service and Industrial Robotics: RAAD 2022*. Springer, 2022, pp. 333–342.
- [TSAI81] Y. Tsai and A. Soni. "Accessible region and synthesis of robot arms". In: (1981).
- [TSAI86] M.-J. Tsai. *Workspace geometric characterization and manipulability of industrial robots*. The Ohio State University, 1986.
- [UEDA01] K. Ueda, I. Hatono, N. Fujii, and J. Vaario. "Line-less Production System Using Self-Organization: A Case Study for BMS". In: *CIRP Annals* 50.1 (2001), pp. 319–322. ISSN: 00078506. DOI: 10.1016/S0007-8506(07)62130-1.
- [UEDA92] K. Ueda. "A concept for bionic manufacturing systems based on DNA-type information". In: *Human Aspects in Computer Integrated Manufacturing*. Elsevier, 1992, pp. 853–863. ISBN: 9780444894656. DOI: 10.1016/B978-0-444-89465-6.50078-8.
- [ULRI81] H. P. D. D. Ulrich. *Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft*. 1981.
- [UR-R10] R. Ur-Rehman, S. Caro, D. Chablat, and P. Wenger. "Multi-objective path placement optimization of parallel kinematics machines based on energy consumption, shaking forces and maximum actuator torques: Application to the Orthoglide". In: *Mechanism and Machine Theory* 45.8 (2010), pp. 1125–1141. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2010.03.008.

- [VAHR12] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann. "Manipulability analysis". In: *2012 12th IEEE International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 568–573.
- [VAHR13] N. Vahrenkamp, T. Asfour, and R. Dillmann. "Robot placement based on reachability inversion". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1970–1975. ISBN: 978-1-4673-5643-5. DOI: 10.1109/ICRA.2013.6630839.
- [VAHR15] N. Vahrenkamp, D. Muth, P. Kaiser, and T. Asfour. "IK-Map: An enhanced workspace representation to support inverse kinematics solvers". In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 785–790. ISBN: 978-1-4799-6885-5. DOI: 10.1109/HUMANOIDS.2015.7363443.
- [VAN 05] J. E. Van Aken. "Management research as a design science: Articulating the research products of mode 2 knowledge production in management". In: *British journal of management* 16.1 (2005), pp. 19–36.
- [VAN 09] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.
- [VAN 96] K. van den Doel and D. K. Pai. "Performance Measures for Robot Manipulators: A Unified Approach". In: *The International Journal of Robotics Research* 15.1 (1996), pp. 92–111. ISSN: 0278-3649. DOI: 10.1177/027836499601500106.
- [VAN 98] H. van Brussel, J. Wyls, P. Valckenaers, L. Bongaerts, and P. Peeters. "Reference architecture for holonic manufacturing systems: PROSA". In: *Computers in Industry* 37.3 (1998), pp. 255–274. ISSN: 01663615. DOI: 10.1016/S0166-3615(98)00102-X.
- [VDI-78] V. S. P. VDI-Gesellschaft Produktion und Logistik and Logistics. *VDI 2815: Begriffe für die Produktionsplanung und -steuerung - Einführung, Grundlagen*. 1978.
- [VERE18] Verein Deutscher Ingenieure, Verband der Elektrotechnik Elektronik Informationstechnik. *Agentensysteme in der Automatisierungstechnik Grundlagen*. Düsseldorf, 2018.
- [VERE21] Verein Deutscher Ingenieure, Verband der Elektrotechnik Elektronik Informationstechnik. *Entwicklung mechatronischer und cyber-physischer Systeme*. Düsseldorf, 2021.
- [VERE90] Verein Deutscher Ingenieure, Verband der Elektrotechnik Elektronik Informationstechnik. *VDI 2860, Montage- und Handhabungstechnik*. Düsseldorf, 1990.
- [VIEI23] L. M. Vieira da Silva, A. Köcher, and A. Fay. "A capability and skill model for heterogeneous autonomous robots". In: *at-Automatisierungstechnik* 71.2 (2023), pp. 140–150.
- [VIJA86] R. Vijaykumar, K. Waldron, and M. Tsai. "Geometric optimization of serial chain manipulator structures for working volume and dexterity". In: *The International Journal of Robotics Research* 5.2 (1986), pp. 91–103.

- [WANG18] B. Wang. "The future of manufacturing: A new perspective". In: *Engineering* 4.5 (2018), pp. 722–728.
- [WECK24] C. Weckendrup, A. Kluge-Wilkes, M. Kweider, A. Göppert, and R. H. Schmitt. "Remanufacturing in der E-Mobilität/Remanufacturing in e-mobility – Adaptive automated battery disassembly using large mobile robots for circular economy". In: *wt Werkstattstechnik online* 114.09 (2024), pp. 482–489. ISSN: 1436-4980. DOI: 10.37544/1436-4980-2024-09-26.
- [WEIN21] T. Weingartshofer, C. Hartl-Nesic, and A. Kugi. "Optimal TCP and robot base placement for a set of complex continuous paths". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 9659–9665.
- [WESE20] M. Weser, J. Bock, S. Schmitt, A. Perzylo, and K. Evers. "An ontology-based metamodel for capability descriptions". In: *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)*. Vol. 1. IEEE. 2020, pp. 1679–1686.
- [WIEN04] H.-P. Wiendahl, D. Gerst, and L. Keunecke, eds. *Variantenbeherrschung in der Montage: Konzept und Praxis der flexiblen Produktionsendstufe*. VDI-Buch. Berlin, Heidelberg and s.l.: Springer Berlin Heidelberg, 2004. ISBN: 978-3-642-18947-0. DOI: 10.1007/978-3-642-18947-0.
- [WIEN07] H.-P. Wiendahl, H. A. ElMaraghy, P. Nyhuis, M. F. Zäh, H.-H. Wiendahl, N. Duffie, and M. Brieke. "Changeable Manufacturing - Classification, Design and Operation". In: *CIRP Annals* 56.2 (2007), pp. 783–809. ISSN: 00078506. DOI: 10.1016/j.cirp.2007.10.003.
- [WOJT20] M. Wojtynek and S. Dr.-Ing. Wrede. *Interactive Workspace Layout focusing on the Reconfiguration with Collaborative Robots in Modular Production Systems: 9-10 December 2020, Online-Event, Germany*. Berlin and Offenbach: VDE VERLAG GMBH, 2020. ISBN: 978-3-8007-5428-1.
- [XIE22] D.-J. Xie, L.-D. Zeng, Z. Xu, S. Guo, G.-H. Cui, and T. Song. "Base position planning of mobile manipulators for assembly tasks in construction environments". In: *Advances in Manufacturing* 11.1 (2022), pp. 93–110.
- [XU20] J. Xu, K. Harada, W. Wan, T. Ueshiba, and Y. Domae. "Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 11018–11024.
- [YOSH85a] T. Yoshikawa. "Dynamic manipulability of robot manipulators". In: *Transactions of the Society of Instrument and Control Engineers* 21.9 (1985), pp. 970–975.
- [YOSH85b] T. Yoshikawa. "Manipulability of Robotic Mechanisms". In: *The International Journal of Robotics Research* 4.2 (1985), pp. 3–9. ISSN: 0278-3649. DOI: 10.1177/027836498500400201.

- [ZACH07] F. Zacharias, C. Borst, and G. Hirzinger. *Capturing Robot Workspace Structure: Representing Robot Capabilities IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007: IROS 2007 ; Oct. 29, 2007 - Nov. 2, 2007, San Diego, CA*. Piscataway, NJ: IEEE Service Center, 2007, pp. 3229–3236. ISBN: 1-4244-0912-8.
- [ZACH08] F. Zacharias, C. Borst, M. Beetz, and G. Hirzinger. “Positioning mobile manipulators to perform constrained linear trajectories”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2578–2584. ISBN: 978-1-4244-2057-5. DOI: 10.1109/IROS.2008.4650617.
- [ZACH09] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger. “Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories”. In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2009, pp. 55–61. ISBN: 978-1-4244-4597-4. DOI: 10.1109/ICHR.2009.5379601.
- [ZACH12] F. Zacharias. *Knowledge representations for planning manipulation tasks*. Vol. 16. Springer Science & Business Media, 2012.
- [ZACH13] F. Zacharias, C. Borst, S. Wolf, and G. Hirzinger. “The capability map: A tool to analyze robot arm workspaces”. In: *International Journal of Humanoid Robotics* 10.04 (2013), p. 1350031.
- [ZHAN17] J. Zhang and X. Fang. “Challenges and key technologies in robotic cell layout design and optimization”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 231.15 (2017), pp. 2912–2924.
- [ZHAN19a] D. Zhang, F. Cursi, and G.-Z. Yang. “Wsrender: A workspace analysis and visualization toolbox for robotic manipulator design and verification”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3836–3843.
- [ZHAN19b] H. Zhang and L. Zhang. “Cloud Robotics Architecture: Trends and Challenges”. In: *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2019, pp. 362–3625. ISBN: 978-1-7281-1442-2. DOI: 10.1109/SOSE.2019.00061.
- [ZHAN19c] J. Zhang, G. Ding, Y. Zou, S. Qin, and J. Fu. “Review of job shop scheduling research and its new perspectives under Industry 4.0”. In: *Journal of Intelligent Manufacturing* 30.4 (2019), pp. 1809–1830.
- [ZHAN20] H. Zhang, Q. Sheng, Y. Sun, X. Sheng, Z. Xiong, and X. Zhu. “A novel coordinated motion planner based on capability map for autonomous mobile manipulator”. In: *Robotics and autonomous systems* 129 (2020), p. 103554.
- [ZHAN22] G. Zhang, F. Du, S. Xue, H. Cheng, X. Zhang, R. Song, and Y. Li. “Design and modeling of a bio-inspired compound continuum robot for minimally invasive surgery”. In: *Machines* 10.6 (2022), p. 468.
- [ZHAN23] X. Zhang, Y. Zhu, Y. Ding, Y. Jiang, Y. Zhu, P. Stone, and S. Zhang. “Symbolic State Space Optimization for Long Horizon Mobile Manipulation Planning”. In: *arXiv preprint arXiv:2307.11889* (2023).

- [ZHU21] K. Zhu and T. Zhang. “Deep reinforcement learning based mobile robot navigation: A review”. In: *Tsinghua Science and Technology* 26.5 (2021), pp. 674–691.
- [ZIMM17] T. A. Zimmerman and T. A. Zimmerman. *Metrics and key performance indicators for robotic cybersecurity performance analysis*. US Department of Commerce, National Institute of Standards and Technology . . . , 2017.
- [ZUMP23] V. Zumper. “Semi-automated Task-based Synthesis and Design of Serial Robotic Systems”. PhD thesis. MECHANISM THEORY, MACHINE DYNAMICS AND ROBOTICS, 2023. DOI: 10.18454/RWTH-2023-02136.

List of related publications

Some of the findings showcased in this thesis have been previously disseminated in articles, proceedings, books, developer platforms or open repositories. The chronological list of these publications is provided below. The author generated all of these publications during their time as a researcher at the Chair of Intelligence in Quality Sensing (IQS) at the WZL of RWTH Aachen University.

1. R. Pugliese, T. Konrad, R. Zweigel, A. Kluge-Wilkes, and R. Schmitt. “Automatische Multikopter-basierte Indoor-Inspektion großer Oberflächen”. In: *Automation 2019*. Ed. by V. W. GmbH. Vol. 2351. VDI-Berichte. Düsseldorf: VDI Verlag, 2019. DOI: <https://doi.org/10.51202/9783181023518-767>. [PUGL19]
Contribution role: Conceptualisation, Methodology, Software, Investigation, Writing - Original Draft
2. A. F. Buckhorst, A. Kluge-Wilkes, and R. H. Schmitt. “Flying Metrology and Defect Identification for Aircraft Surface Inspection: Automated optical investigation of aircraft surfaces after lightning strike events”. In: *PhotonicsViews* 16.1 (2019), pp. 68–71. DOI: <https://doi.org/10.1002/phvs.201900009>. [BUCK19a]
Contribution role: Conceptualisation, Methodology, Investigation, Writing - Original Draft
3. C. Nicksch, A. Kluge-Wilkes, M. Huber, and R. H. Schmitt. “Global Reference System for factory-wide integration of metrology enabling flexible automation in aeroplane assembly-requirements, concept and suitable technologies”. In: *Procedia Manufacturing* 52 (2020), pp. 89–94. DOI: <https://doi.org/10.1016/j.promfg.2020.11.017>, . [NICK20]
Contribution role: Conceptualisation, Methodology, Investigation, Writing - Original Draft
4. A. Kluge-Wilkes and R. H. Schmitt. “Dynamic Task Allocation for Cooperating, Heterogeneous Assembly Resources in LMAS”. in: *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems*. Ed. by A.-L. Andersen, R. Andersen, T. D. Brunoe, M. S. S. Larsen, K. Nielsen, A. Napoleone, and S. Kjeldgaard. Springer eBook Collection. Cham: Springer International Publishing and Imprint Springer, 2022, pp. 535–542. ISBN: 978-3-030-90700-6. DOI: https://doi.org/10.1007/978-3-030-90700-6_60. [KLUG22c]
Contribution role: Conceptualisation, Methodology, Software, Writing - Original Draft
5. A. Kluge-Wilkes and R. H. Schmitt. “Mobile Robot Base Placement for Assembly Systems: Survey, Measures and Task Clustering”. In: *Production at the Leading Edge of Technology*. Ed. by B.-A. Behrens, A. Brosius, W.-G. Drossel, W. Hintze, S. Ihlenfeldt, and P. Nyhuis. Lecture Notes in Production Engineering. Cham: Springer International Publishing, 2022, pp. 439–447. ISBN: 978-3-030-78423-2. DOI: [10.1007/978-3-030-78424-9_49](https://doi.org/10.1007/978-3-030-78424-9_49). [KLUG22d]
Contribution role: Conceptualisation, Methodology, Software, Validation, Writing - Original Draft, Visualisation

6. A. Kluge-Wilkes, B. Gunaseelan, and R. H. Schmitt. "Ontology-based task allocation for heterogeneous resources in line-less mobile assembly systems". In: *MHI Colloquium*. Springer. 2022, pp. 53–63. DOI: https://doi.org/10.1007/978-3-031-10071-0_5. [KLUG22b]
Contribution role: Conceptualisation, Methodology, Software, Validation, Writing - Original Draft, Visualisation
7. Code: A. Kluge-Wilkes and B. Gunaseelan. *Ontology-based task allocation for heterogeneous resources in Line-less Mobile Assembly Systems*. https://git-ce.rwth-aachen.de/wzl-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/capilano. 2022. [KLUG22a]
Contribution role: Conceptualisation, Methodology, Software
8. A. Kluge-Wilkes, R. Baier, I. Kunze, A. Müller, A. Shahidi, D. Wolfschläger, C. Brecher, B. Corves, M. Hüsing, V. Nitsch, R. H. Schmitt, and K. Wehrl. "Modular Control and Services to Operate Lineless Mobile Assembly Systems". In: *Internet of Production: Fundamentals, Applications and Proceedings*. Ed. by C. Brecher, G. Schuh, W. van der Aalst, M. Jarke, F. T. Piller, and M. Padberg. Cham: Springer International Publishing, 2023, pp. 1–26. ISBN: 978-3-030-98062-7. DOI: [10.1007/978-3-030-98062-7_13-2](https://doi.org/10.1007/978-3-030-98062-7_13-2). [KLUG23d]
Contribution role: Conceptualisation, Methodology, Software, Writing - Original Draft, Visualisation
9. A. Kluge-Wilkes, A. Göppert, and R. H. Schmitt. "End-Effector-Dependent Evaluation of Task Executability by Mobile Robots in Changeable Assembly Stations". In: *Proceedings of the Changeable, Agile, Reconfigurable and Virtual Production Conference and the World Mass Customization & Personalization Conference*. Springer. 2023, pp. 536–543. DOI: https://doi.org/10.1007/978-3-031-34821-1_58. [KLUG23e]
Contribution role: Conceptualisation, Methodology, Software, Validation, Writing - Original Draft, Visualisation
10. Code: A. Kluge-Wilkes. *End-effector dependent executability*. https://git-ce.rwth-aachen.de/wzl-mq-public/iop/ws-b2.iv_formation-planning-of-mobile-robots/end-effector-dependent-executability. 2023. [KLUG23b]
Contribution role: Conceptualisation, Methodology, Software, Validation, Writing - Original Draft, Visualisation
11. M. Behery, P. Brauner, A. Kluge-Wilkes, R. Baier, A. Mertens, R. H. Schmitt, M. Ziefle, and G. Lakemeyer. "Digital Shadows for Robotic Assembly in the World Wide Lab". In: *Procedia CIRP* 120 (2023). 56th CIRP International Conference on Manufacturing Systems 2023, pp. 165–170. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2023.08.030>. [BEHE23]
Contribution role: Conceptualisation, Methodology, Software
12. A. Kluge-Wilkes, P. D. Reverdito, S. Kohar, A. Göppert, and R. H. Schmitt. "Evaluating task executability of mobile robots with performance maps". In: *Procedia CIRP* 127 (2024). 10th CIRP Conference on Assembly Technology and Systems (CIRP CATS 2024), pp. 38–43. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2024.07>.

008. [KLUG24b]
Contribution role: Conceptualisation, Methodology, Software, Validation, Writing - Original Draft, Visualisation
13. Data: A. Kluge-Wilkes. *Extending the executability of assembly task poses by robot through end-effectors*. Version 1.0. Zenodo, Dec. 2023. DOI: 10.5281/zenodo.10119467. [KLUG23c]
Contribution role: Data Curation, Software
 14. Data: A. Kluge-Wilkes. *Evaluating robot workspaces with performance maps*. Zenodo, Dec. 2024. DOI: <https://zenodo.org/records/10462326>. [KLUG24a]
Contribution role: Data Curation, Software
 15. C. Weckendrup, A. Kluge-Wilkes, M. Kweider, A. Göppert, and R. H. Schmitt. "Remanufacturing in der E-Mobilität/Remanufacturing in e-mobility – Adaptive automated battery disassembly using large mobile robots for circular economy". In: *wt Werkstattstechnik online* 114.09 (2024), pp. 482–489. ISSN: 1436-4980. DOI: 10.37544/1436-4980-2024-09-26. [WECK24]
Contribution role: Conceptualisation, Writing - Review & Editing
 16. A. Avhad, C. Schou, A. Kluge-Wilkes, A. Göppert, R. H. Schmitt, and O. Madsen. "A Reference Architecture for Swarm Production: A Top-down Approach Supported by Holonic ARTI". in: *Journal of Industrial Information Integration* (2024). (In review). [AVHA24]
Contribution role: Methodology, Writing - Review & Editing
 17. A. Göppert, C. Schou, A. Kluge-Wilkes, L. Bergs, I. Rodriguez, L. Schäper, H. Arnason, D. Arias-Cachero Rincon, N. König, O. Madsen, P. Mogensen, and R. H. Schmitt. *5G Edge-Cloud-based Robotics for Flexible and Resilient Production Systems*. (In review). 2024. [GÖPP24]
Contribution role: Conceptualisation, Writing - Original Draft, Visualisation

List of contributing theses

The author had the privilege of supervising the following theses during her time researching the topic of this dissertation at the Chair of Intelligence in Quality Sensing at the WZL of RWTH Aachen University. The author wishes to extend heartfelt appreciation for the dedication shown by these thesis students.

Table 9.1: List of contributing theses. BT = bachelor thesis, MT = master thesis.

Name	Title	Contribution	Type	Year
Sard Kosova	Werkzeugabhängige Erreichbarkeits- und Fähigkeitsbeschreibung mobiler Montageroboter	Ch. 4	MT	2021
Balaji Gunaseelan	Ontology-based decision framework for capability-based task allocation in Line-less Mobile Assembly Systems (LMAS)	Ch. 3	MT	2022
Chenyang Jiang, Mikolaj Radlinski, Van Dao, Jannick Vitak	Werkzeugabhängigen Erreichbarkeits- und Fähigkeitsbeschreibung mobiler Montageroboter	Ch. 4	SPP - Software Projektpraktikum	2022
Michael Philipp	Bewertung von Formationen mobiler Montageroboter anhand der Schnittmengen von 3-D Volumen.	Ch. 5	BT	2022
Ege Balkan	Autonomous Clustering-based Task Allocation and Base Placement for Mobile Robots in Line-less Mobile Assembly Systems	Ch. 5	MT	2023
Stefanus Kohar	Workspace evaluation to place mobile robots in changeable assembly stations	Ch. 4	MT	2023

A Motivation, research methodology and structure

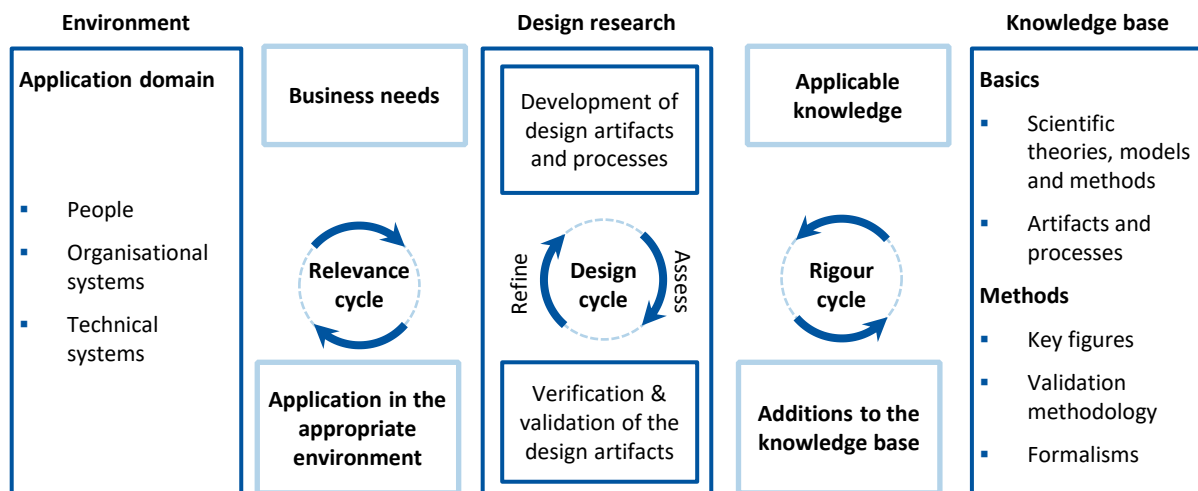


Figure A.1: Design Science Research framework, based on [HEVN07, p.88; HEVN04].

B Methodological approach to the systematic literature review

Table B.1: Search strings and keywords for the systematic literature mapping and review.

Number	Research domain	Search string
1	Formation planning	formation $\cap$ planning $\cap$ mobile $\cap$ assembly $\cap$ robots
2	Review / survey	(review $\cup$ survey) $\cap$ (assembly $\cap$ (robots $\cup$ manipulators))
3	Robot cell layout and design	((design $\cup$ layout) $\cup$ planning) $\cap$ robot $\cap$ (cell $\cup$ (assembly $\cap$ station))
4	Metrics for formation planning	(metrics $\cup$ measures $\cup$ (key $\cap$ figures) $\cup$ evaluation) $\cap$ (((robot $\cup$ manipulators) $\cap$ assembly) $\cup$ (robot $\cap$ (cells $\cup$ (assembly $\cap$ station))))
5	Selecting robots for assembly	((assembly $\cap$ resource) $\cup$ robot) $\cap$ (selection $\cup$ identification $\cup$ determination) $\cap$ ((assembly $\cap$ tasks) $\cup$ capabilities)
6	Task clustering or task allocation	task $\cap$ (clustering $\cup$ allocation $\cup$ assignment) $\cap$ ((assembly $\cap$ (station $\cup$ robot $\cup$ manipulator)) $\cup$ (robot $\cap$ cell))
7	Robot base placement / placement of workpiece	(assembly $\cap$ (robot $\cup$ manipulators)) $\cap$ ((base $\cup$ workpiece) $\cap$ placement $\cap$ optimisation) $\cup$ positioning $\cup$ (layout $\cap$ planning) $\cup$ configuration $\cup$ (posture $\cap$ optimisation))

Table B.2: Overview of the retrieved results (excluding replicas) after the phases of paper identification and screening of the literature mapping and review.

Nr.	Research domain	Number of retrieved results			Number of selected results	
		Scopus (titles)	Scopus (abstracts)	Research Gate	after title screening	after ab- stract screening
1	Formation plan- ning	0	7	1	1	0
2	Review / survey	7	352	147	11	2
3	Robot cell layout and design	185	1000	500	157	34
4	Metrics for forma- tion planning	65	1000	439	52	7
5	Selecting robots for assembly	6	1000	500	56	2
6	Task clustering or task allocation	30	467	356	40	6
7	Robot base place- ment / placement of workpiece	67	1000	500	61	17
Sum		360	4826	2443	378	68
			7629			

C Ontology-based definition

Figure C.1 visualises the syntax of the class models of CAPILANO in UML notation. The ontology consists of the classes of the **processes** — defining the assembly sequence of the process steps, **products** — consisting of the number of parts and components, **resources** — constituting physical and cyber instances of the factory, and **capabilities**, depicted in dark gray. Each assembly resource is represented as an instance of a component class. Following SOIL and AAS each instance has a type, a state and specific properties. As can be seen, capabilities are the central connections between the process steps and the resources. A capability enables the transition between the process step states by connecting parts to products. The process steps comprise the entire assembly process, which aims to combine different parts into a finished product. Thereby, the assembly sequence defines the dependencies and order of the process steps by providing a precedence graph of the process steps. The process steps refer to a required capability, which may be an elemental capability or a composed capability consisting of several elemental capabilities — for a definition of elemental and composed capabilities, see section 4.3.2. Moreover, the process steps give insight into the duration of operation to execute the given process step. The different resources provide the required capabilities to execute the process steps. As resources, one can differentiate between physical resources, like robots, humans, sensors and end-effectors, which all provide different capabilities in dependence on their combination with each other. A factory consists of assembly stations, and the station consists of physical and cyber resources. Cyber resources, like computing power, methods and algorithms, are not further considered here but generally can be included in the developed schema of instances in CAPILANO.

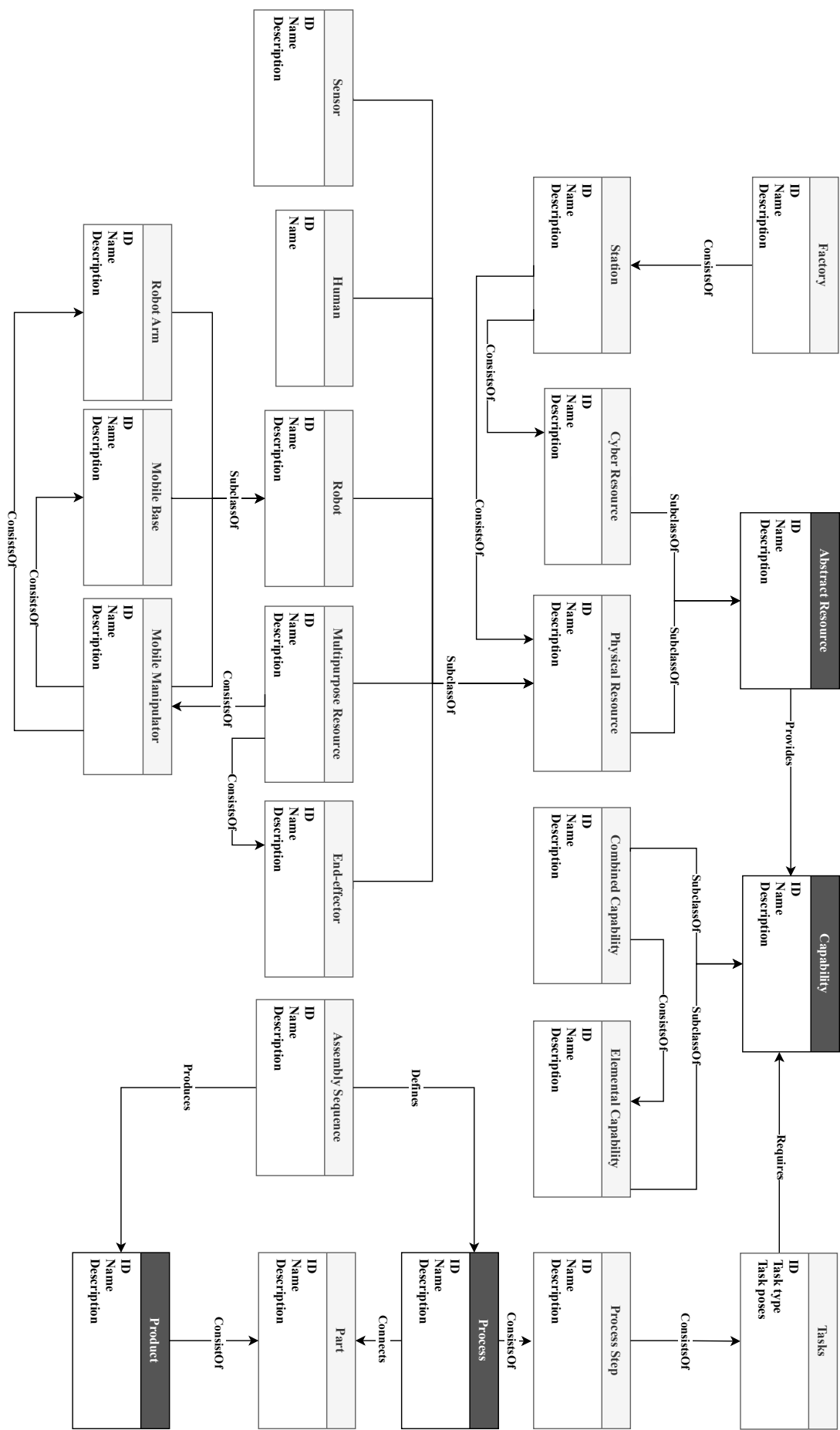


Figure C.1: CAPILANO — class diagram.

D Workspace evaluation of mobile robots in assembly

A concise literature review was conducted to get an overview of performance metrics for evaluating robots during the robot design phase, during operation or as indicated on the robot specification sheets.

Table D.1: Overview and classification of robot performance metrics.

Begin of table D.1					
Metric	Explanation and source	Range: local (L) or global (G)	Physical reference: kinematic (K) or dynamic (D)	Task reference: intrinsic (I) or extrinsic (E)	Scale: absolute (A) or relative (R)
Reach	[m]	G	K	I	A
Payload or maximum load	[kg]	G	D	I	A
Weight	[kg]	G	D	I	A
Wrist torque ³⁷	[Nm]	G	K	I/E	A
Accuracy ³⁷	[m] Mean / maximum error	G	K	I/E	A
Repeatability ³⁷ (pose or path)	[m] Mean / maximum error deviation	G	K	I/E	A
Degrees of Freedom (DoF)	[—]	G	—	I	A
Number of Axes	[—]	G	—	I	A
Power consumption	[V,Hz]	G	D	E	A
Maximum axis speed / velocity ³⁷	$\begin{bmatrix} \circ \\ s \end{bmatrix}$	G	K	I/E	A
Joint limits ³⁷	[°]	G	D	I/E	A
Job execution time ³⁷	[s]	G	K/ D	I/E	A
Workspace volume V	Reachable volume (equivalent to the workspace itself) $V = \int_x dx$ with the robot flange position x [PATE15, p.557] or as denoted in equation 2.1 by [ZHAN19b]	G	K	I	A

Continuation of table D.1							
Metric		Explanation	Range	Physics	Task	Scale	
Workspace index (WSI)		$WSI = \frac{n_{ws}}{n_{obj}} (n_{ws} \leq n_{obj})$ Percentage of the reachable objective workspace, [PUGL12, p.1040]	L	K	I	R	[0,1]
Minimum singular value		Indicates closeness to singularities [TANE00, p.70]	L	K	I	R	$(0, \infty]$
Condition number κ		Kinematic isotropy of the Jacobian $\kappa = J J^{-1} $ [GOSS91, p.113]	L	K	I	R	$(0, \infty]$ [TANE00, p.69]
Local conditioning index LCI (sometimes: Reciprocal condition number c_r)		In difference to the conditioning number, the local condition index has an upper limit $LCI = \frac{1}{\kappa}$ [TANE00, p.69]	L	K	I	R	$(0, 1]$
Global conditioning index η		Global weighted average of the conditioning index	G	K	I	R	$(0, 1)$
Manipulability measure w		Distance to singular values. Measures "how easy" it is for the robot to move out of the current pose (see Eq. D.2) [YOSH85b]	L	K	I	R	
Extended manipulability measure		Extension of manipulability, by consideration of joint limits, redundancy and obstacles: Closeness to joint limits or obstacles is penalised, redundancy of configuration solutions per pose is rewarded [VAHR12]	L	K	I	R	
Global manipulability measure w_d		Global integral of manipulability measure; $GMI = \frac{\int_W (w) dW}{\int_W dW}$ with W specific point in workspace	G	K	I	R	
Dexterity index D		$D = \frac{1}{3} * (\frac{\Delta\gamma}{2*\pi} + \frac{\Delta\beta}{2*\pi} + \frac{\Delta\alpha}{2*\pi})$ [TANE00] with $\Delta\alpha$ etc. range of orientation of manipulator. Calculated for variations of α, β and γ according to the algorithm proposed by [TANE00]	L	K	I	R	
Reachability index D_{RI}		The reachability index D_{RI} is calculated for discrete positions in the workspace and indicates the percentage of orientations that can be reached from these positions. [ZACH13, p.10]	L	K	I	R	$[0,1]$
Mean dexterity index D_{Mean}		Mean of dexterity indices over a given region: $D_{Mean} = \frac{\sum D}{N}$ [PATE15] with D is the dexterity index of a location, and N is the number of locations taken into account	L/G	K	I	R	

Continuation of table D.1

Metric	Explanation	Range	Physics	Task	Scale
Joint range availability JRA	Joint displacement relative to maximal joint displacement [KLEI87, p.78] $JRA = \sum (\theta_i - \theta_{ci})^2$ or $JRA = \sum (\frac{\theta_i - \theta_{ci}}{\theta_{i,max}})^2$ [KLEI87, p.78]	L	K	I	R
Joint mid-range proximity z	Relative proximity to the middle of the joint range $z = \frac{1}{2}(\theta - \bar{\theta})TW(\theta - \bar{\theta})$ with $\bar{\theta} = \frac{1}{2}(\theta_{min} + \theta_{max})$ and W pos.def. [VAHR12] [TSAI86]	L	K	I	R
Service angle/service sphere $SA(P)$	Local dexterity measure; $SA(P) = \bigcup_{i=1}^{NR} \frac{A(SR)_i}{4\pi h^2}$ [YOSH85a] where $SA(P)$ is service angle of point P , N_R is the number of service regions, $A(SR)_i$ is the service angle of the i th service region and h is the length of the manipulators hand, "service angle is defined as the total range of the approach angle of a manipulator around a point of the workspace" [TANE00, p.1]	L	K	I	R
Global transmission index	Measures how efficiently an input is transformed into an output [CHEN07]	G	D	I	R
Stiffness matrix	Symmetric, semi- or positive-definite matrix representing the potential energy in a mechanical system [KÖVE07]	L	D	I	R
Dynamic manipulability measure w_d	Extension of the manipulability measure to reflect the dynamic properties; $w_d = \sqrt{\det[J(M.M^T)^{-1}J^T]}$ or $w_d = \frac{\det(J)}{\det(M)} $ [YOSH85a]	L	D	I	R
Isotropic index Δ	Measures how even the manipulability measure is; $\Delta = \frac{\mu_o}{\Psi}$	L	D	I	R
Joint velocity measure	Compares current and needed joint angle $\Psi_{JVM} = \sum (\theta_i - \theta_{i,curr})^2$	L	K	E	R
Manipulator velocity ration r_v	Relation between end-effector velocity and joint velocity; $r_v = \sqrt{\frac{\dot{x}_v^T \dot{x}_v}{\dot{\theta}_v^T \dot{\theta}_v}}$ with $\dot{X}_v = \sqrt{W_X} \dot{X}$	L	K/D	I	R
Energy consumption	Varying [RUSS22, p.7]	G	D	E	A
Workspace evaluation measure G	Combines the percentage of free workspace area (not occupied by objects), the percentage of reachable objects in the workspace and the average manipulability [WOJT20, p.188]	G	K	E	R [0,1]

Continuation of table D.1				
Metric	Explanation	Range	PhysicsTask	Scale
Approximation of the minimum manipulator velocity ratio $AMMVR$	Calculated using a finite set of manipulator velocity ratio values computed at specific points along a path chosen by a trajectory approximation algorithm. [NEKT10]	L	D	E
End of table D.1				

Manipulability

YOSHIKAWA introduced the local "**manipulability measure**" w (sometimes manipulability index) in 1985. The manipulability measure provides information about the distance of the current robot configuration to a singular configuration by computing the ellipsoid of the singular vectors of the Jacobian. [YOSH85b]

The **Jacobian** $J(\theta)$ maps the robot flange velocity $\dot{x}$ in Euclidean space to the joint velocity vector $\dot{\theta}$, with $x = (x, y, z, \alpha, \beta, \gamma)^T$ as the pose of the robot flange. For a robot with n DoFs and therefore n joint variables $\theta_i, i = 1, 2, \dots, n$ and the joint vector $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$, it follows:

$$\dot{x} = J(\theta)\dot{\theta} \quad (D.1)$$

If the rank of the Jacobian is lower than the dimension of the task m (e.g. three-dimensional or six-dimensional etc.) $rank J(\theta) < m$, a *singularity* of the robot occurs. Thus, when the determinant of the Jacobian approaches zero, the robot approaches a singularity [TANE00, p.6].

With this, the manipulability measure w is defined as follows:

$$w = \sqrt{\det J(\theta) J^T(\theta)} \quad (D.2)$$

$$w = \sigma_1 * \sigma_2 \dots \sigma_n = \prod_{i=1}^n \sigma_i \quad (D.3)$$

with the singular values $\sigma_1, \sigma_2 \dots \sigma_n$ of the Jacobian.

As only non-redundant industrial robots with six DoFs are considered here and for all non-redundant robots, the Jacobian is a square matrix, the calculation can be reduced to:

$$w = |\det J(\theta)| \quad (D.4)$$

³⁷These measures are typically given as intrinsic, global parameters in commercial data sheets (indicating the maximum safe value). At the same time, National Institute of Standards and Technology (NIST) handles them as local measures, and during operation, they might be calculated locally and extrinsic. Zimmerman and Zimmerman

Here, all assembly tasks are described as a six DoFs pose. Therefore, $m = n$ follows. Moreover, (as is true for all PUMA-robots), w is not related to the angular value of the first axis [YOSH85b, p.6].

The scalar w can be interpreted as the ability of the robot to change from one specific configuration into another [MALH19a, p.5585]. The manipulability measure is proportional to the realisable velocity of the robot flange from a specific configuration on [YOSH85b, p.4]. The manipulability measure measures kinematic dexterity by considering the end-effector motion in all directions. The higher w , the further away a singularity is. Since w becomes equal to zero at all singularities, no distinction can be made between different singularities [CHAN88]. The Jacobian and, therefore, w can be computed analytically based on the joint angles.

YOSHIKAWA 1985 extended the manipulability measures to incorporate dynamic effects by including the inertia matrix (M). The **dynamic manipulability index** quantifies the degree of arbitrariness of acceleration change ($\dot{\hat{v}}$) under the constraint of the joint driving force ($\dot{\hat{\tau}}$). The relationship is described as follows:

$$\dot{\hat{v}} = JM^{-1}\dot{\hat{\tau}} \quad (D.5)$$

from which the dynamic manipulability index follows:

$$w_d = \sqrt{\det[J(M^T M)^{-1}J^T]} \quad (D.6)$$

Similarly, the index can also be calculated through the singular values of the matrix JM^{-1}

$$w_d = \prod_{i=1}^m \sigma_i \quad (D.7)$$

for the non-redundant robots, the formula is simplified to

$$w_d = \left| \frac{\det J}{\det M} \right| \quad (D.8)$$

The manipulability measure is also a good metric of kinematic dexterity as it considers the EE motion in all directions. The preferred value of this index is closer to unity (1), with values ranging from 1 to ∞

The manipulability index suffers from several limitations

- **Scale Dependency**, the selection of physical units within the Jacobian will influence the value of the Index. Therefore, the index will have different values in different units representing the Jacobian.
- **Dimensional Dependency**, the manipulator configuration determines the dimension units used in the Jacobian. A combination of prismatic and revolute joints would produce a non-homogeneous Jacobian. As such, a consistent value of the determinant and singular of the Jacobian cannot be achieved.

- **Unbounded Index**, the value of the manipulability index does not lie between a defined range. A comparison between two different manipulators is not feasible. Therefore, it will only become a relative index by comparison with other indexes within the same workspace.
- **Order Dependency**, increasing manipulator links will increase the order of the index. Comparison of two manipulators with different numbers of links is not feasible, as the indexes have different orders.

VAHRENKAMP ET AL. 2012 propose the extended manipulability measure, which penalises solutions close to joint limits or solutions close to areas where collisions might occur. Thereby, additional constraints are considered. [VAHR12, p.567]

Precision

The condition number κ indicates the kinematic isotropy of the Jacobian and is interpretable as a "measure of directional uniformity of the manipulability ellipsoid" [sic! ellipsoid] [YOSH85b, p.8]. Additionally, it provides information about linear independence of the columns of the Jacobian [TANE00, p.69] and "can be regarded as a measure of the Jacobian invertibility" [GOSS91, p.220].

$$\kappa = \|J\| \|J^{-1}\| \quad (\text{D.9})$$

TANEV ET AL. proposed the **local condition index** through the reciprocal of the condition number to provide a bounded metric.

$$LCI = \frac{1}{\kappa} \quad (\text{D.10})$$

Providing a global condition index, several calculation methods have been proposed. The **kinematic conditioning index** proposed by ANGELES ET AL. [ANGE92] indicates the worst performance point in the whole workspace

$$KCI = \frac{1}{\kappa_{min}} * 100\% \quad (\text{D.11})$$

STOCCO ET AL. proposed the **global isotropy index** to represent the worst performance of the manipulator over the entire workspace:

$$GII = \frac{\sigma_{WS_{min}}}{\sigma_{WS_{max}}} \quad (\text{D.12})$$

With the $\sigma_{WS_{min}}$ as the minimum and $\sigma_{WS_{max}}$ as the maximum singular value over the entire workspace.

While the **global conditioning index** gives the average of the condition number κ for the workspace volume V in Cartesian space:

$$\eta = \frac{\int_x (\kappa) dx}{\int_x dx} \quad (D.13)$$

with x as a specific point in the workspace. The global conditioning index formulated in joint space:

$$\eta = \frac{\int_R \left| \left(\frac{1}{\kappa} \right)^2 \Delta \right| d\theta_n \dots d\theta_2 d\theta_1}{\int_R |\Delta| d\theta_n \dots d\theta_2 d\theta_1} \quad (D.14)$$

with R as the workspace in joint space and Δ as the determinant of the Jacobian $\det(J)$. [GOSS91, p.221]

Providing information about the isotropy KIM ET AL. proposed the **isotropic index** by combining the manipulability index and the eigenvalues:

$$\Delta = \frac{w_O}{\Psi} \quad (D.15)$$

where Ψ is the arithmetic mean of the eigenvalues and w_O is the order-independent manipulability index.

Dexterity

KUMAR ET AL. 1981 propose the "**dexterity index**" D as "a measure of a manipulator to achieve different orientations for each point within the workspace" [TANE00, p.65]. Any position can be reached in the volume of a "reachable" workspace, while in the "dextrous" workspace, any pose can be reached. [KUMA81] Locally, the dexterity D quantifies the capability of reaching varying orientations in a single position [PATE15, p.551].

$$D = \frac{1}{3} * \left(\frac{\Delta\gamma}{2 * \pi} + \frac{\Delta\beta}{2 * \pi} + \frac{\Delta\alpha}{2 * \pi} \right) \quad (D.16)$$

with Δ as the achievable range of the rotation angles roll α , pitch β and yaw γ in relation to the respective full range of 0° to 360° of α, β and γ [TANE00, p.65]. If $dx = \frac{\Delta\alpha}{2\pi} = 1$, then we talk about full dexterity. In a position from which multiple orientations are achievable, the dexterity index will have higher values than in positions with a single orientation solution.

The index is averaged over the workspace considering N local values to convert it into a global metric.

$$D_{mean} = \frac{\sum D}{N} \quad (D.17)$$

Zacharias et al. 2007 and Makhal et al. 2018 provide a numerical approach of calculating the "Reachability Index" D_{RI} by considering n orientations per position and giving out the percentage of how many of these tested orientations are reachable (R).

$$D_{RI} = \frac{R}{n} \cdot 100 \quad (D.18)$$

Others

TSAI'S **penalty function** $p(\theta)$ penalises configurations near joint limits but neglects well-reachable poses near limits due to redundant configurations [TSAI86]. HUO ET AL. [HUO08] extended the idea, by proposing the joint mid-range proximity index. They penalised configuration closer to the joint limits, as a condition for an objective function for a manipulator control algorithm.

Extrinsic metrics compare the achievable metric and the known goal value during task execution. CHIU compare the desired task and achievable manipulator transmission ratio. LEE optimise the achievable static force application in a given position. STURGES quantify the difficulty based on the dexterity of the task-manipulator interaction. Extrinsic metrics require a defined task description and known goal values during task execution. Therefore, extrinsic metrics are only applicable if the condition of a goal manipulator trajectory with quantified metrics is met.

Applicability of performance metrics

Jacobian-based metrics are applied for design and topology planning Angeles et al., workspace optimisation, robot arm path planning [KLEI87] and control [RUSS22]. The utilisation of the Jacobian brings the following limitations regarding applicability: The Jacobian depends on the utilised scale (dimensions/units/size e.g. m, mm, inch) [TANE00, p.67], [RUSS22, p.10]. Thus, values differ, and the scale has to be given to ensure reproducibility and comparability between different values.

The manipulability measure is only influenced by the positioning of the links and the robot's structure. Another disadvantage of performance criteria based on the eigenvalues of the Jacobian matrix is the dimensionality dependence due to the inhomogeneity of the Jacobian matrix when combining translational and rotational components.

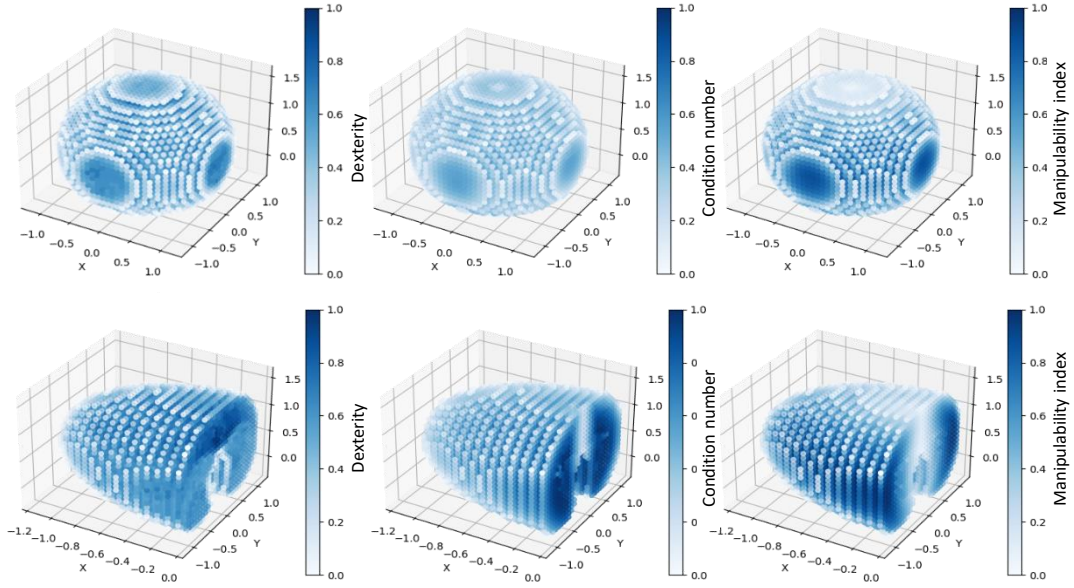
E Evaluation of the executability of assembly tasks by a robot

VDI 2860 represents tasks by combining solution-neutral elementary and sub-functions [VERE90]. Therein, handling tasks are defined as creating, modifying, or temporarily maintaining geometrically determined bodies for a given spatial arrangement ($\hat{=}$ pose as the combination of position and orientation) in a reference coordinate system. In VDI 2860, manipulating, transporting and storing are listed as the part-functions of material flow. During manipulation, all six DoFs have to be defined, whereas for transporting and storing, only the start and end positions are relevant (three DoFs). Handling is the creation, defined alteration or temporary upholding of a spatial arrangement of geometrically determined bodies in a reference coordinate system. [VERE90, p.2–3]

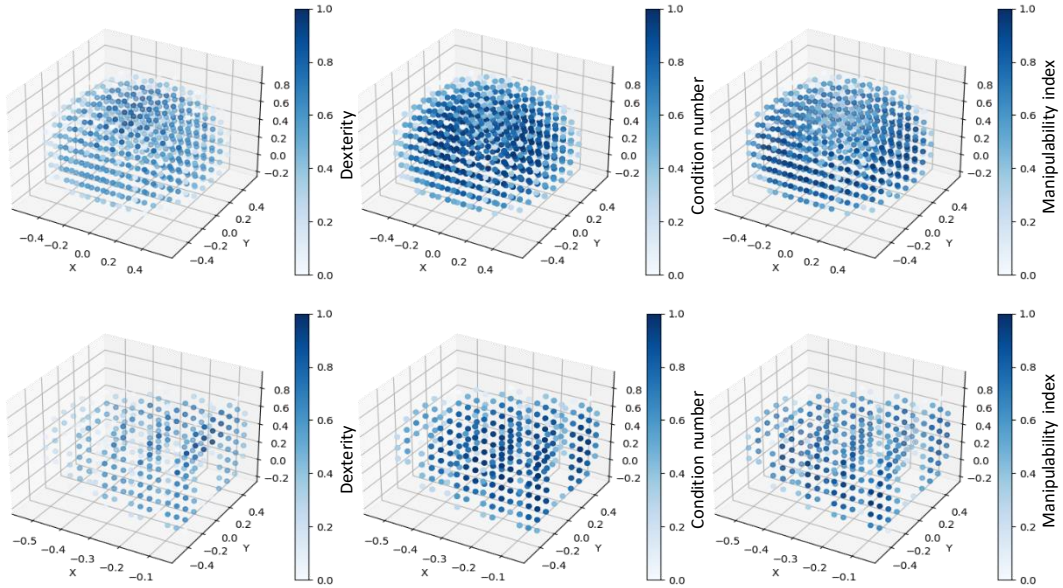
Table E.1: Overview and classification of robot capabilities.

Begin of table E.1					
Process type	Process step	Capability	Movement: Discrete (D) or Continuous (C)	Precision: High (H) or Low (L)	TCP range: Reachable (R) or Dexterous (DX)
Assembling [DEUT03]	Putting together [DEUT22, p.13]	Inserting	D	H	DX
		Nesting	D	H	DX
		Laying in, Laying on	D	H	DX
		Hanging into something	D	L	R
	Filling		D	L	R
	Pressing (in)	Screwing	D	H	DX
		Clamping	D	H	DX
		Pressure joining	D	H	DX
	Joining by forming [DEUT22, p.11]	Pressure forming (Freeforming, Mould- ing, Pushing through)	D	H	R
		Tensile pressure forming (Pushing)	D	L	R
		Bending with active energy	C	H	DX
		Thrust forming (Twisting: shear forming with rotary tool movement)	D	H	R

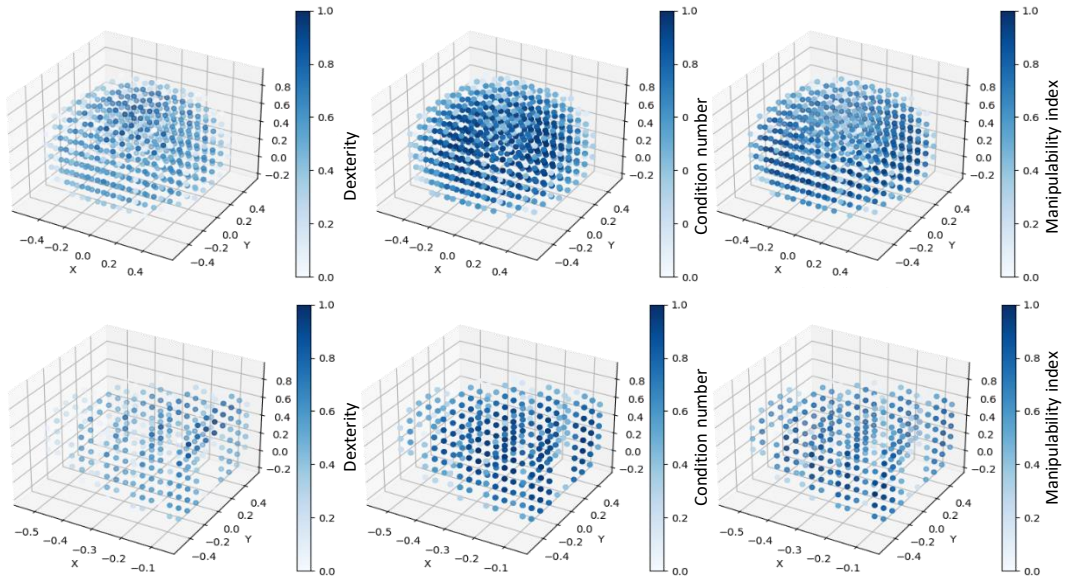
Continuation of table E.1						
Task type	Task		Capability	Movement	Precision	Range
	Welding	[DEUT22, p.13]	Pressure welding, Fusion welding	C	H	R
	Soldering	[DEUT22, p.13]	Soft soldering, Brazing	C	H	R
	Gluing	[DEUT22, p.13]	Gluing with physically or chemically setting adhesives	C	H	R
	Storing			D	L	R
Handling [VERE90]	Transporting			D	L	R
	Manipulating			C	L	R
	Controlling	[DEUT95, p.6]	Inspecting	C	L	R
			Measuring	D	H	R
Separating [DEUT22]	Machining with geometrically defined cutting edges		Drilling [DEUT22, p.12]	D	H	DX
	Machining with geometrically defined cutting edges		Filing, rasping [DEUT22, p.12]	C	L	R
	Disassembling [DEUT22, p.12]		Removing submodules	C	L	R
			Draining, removing	D	L	R
			Loosening friction-locked connections	C	L	R
			Loosening of adhesive connections	C	L	R
Coating [DEUT22; DEUT03]	Painting	[DEUT22, p.14]		C	L	DX
	Lacquering	[DEUT22, p.14]		C	L	DX
	Colouring	[DEUT22, p.14]		C	L	DX
	Printing, Marking			C	H	R
	Labelling			C	L	R
	Spatulaing			C	L	R
	Plastering			C	L	R
	Coating by welding			C	H	R
	Coating by soldering			C	H	R
End of table E.1						



a) Run order 1: 80mm resolution, 25 poses per voxel, robot model ABB IRB 1600-1.2

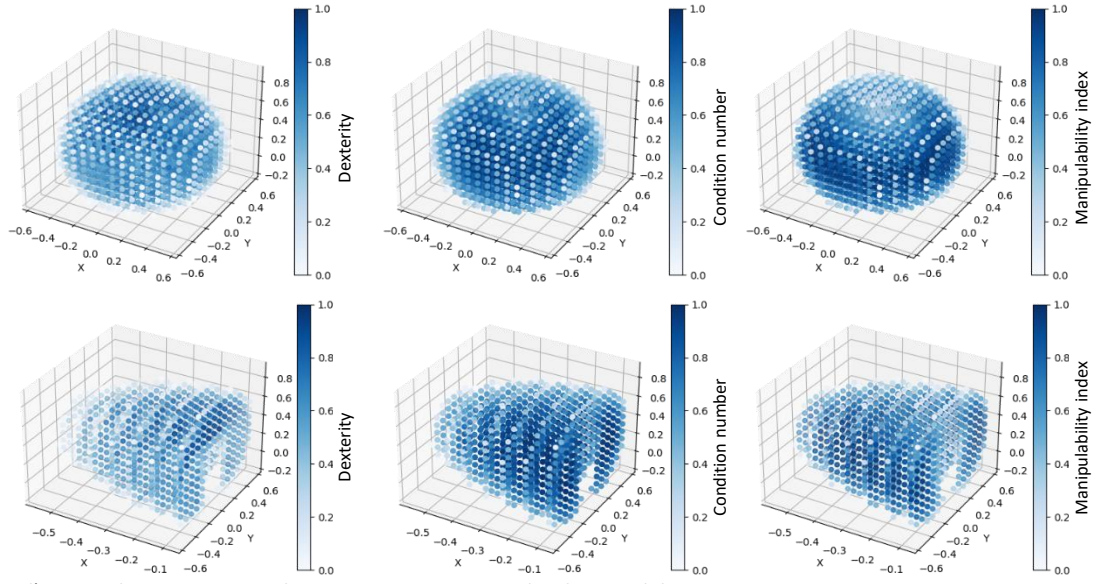


b) Run order 2: 120mm resolution, 25 poses per voxel, robot model ABB IRB 120-0.58

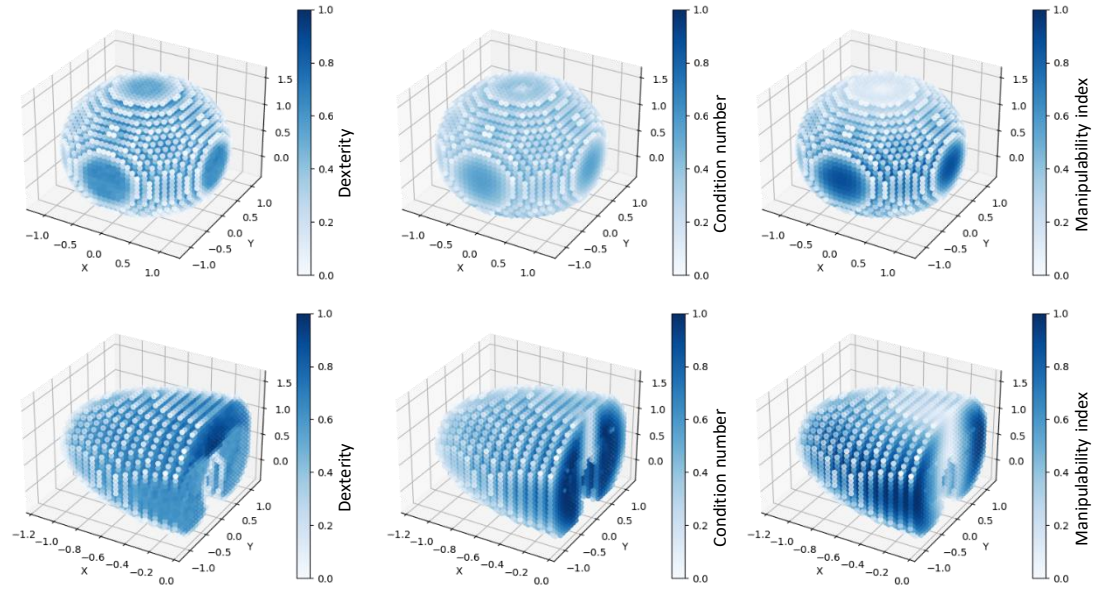


c) Run order 3: 120mm resolution, 50 poses per voxel, robot model ABB IRB 120-0.58

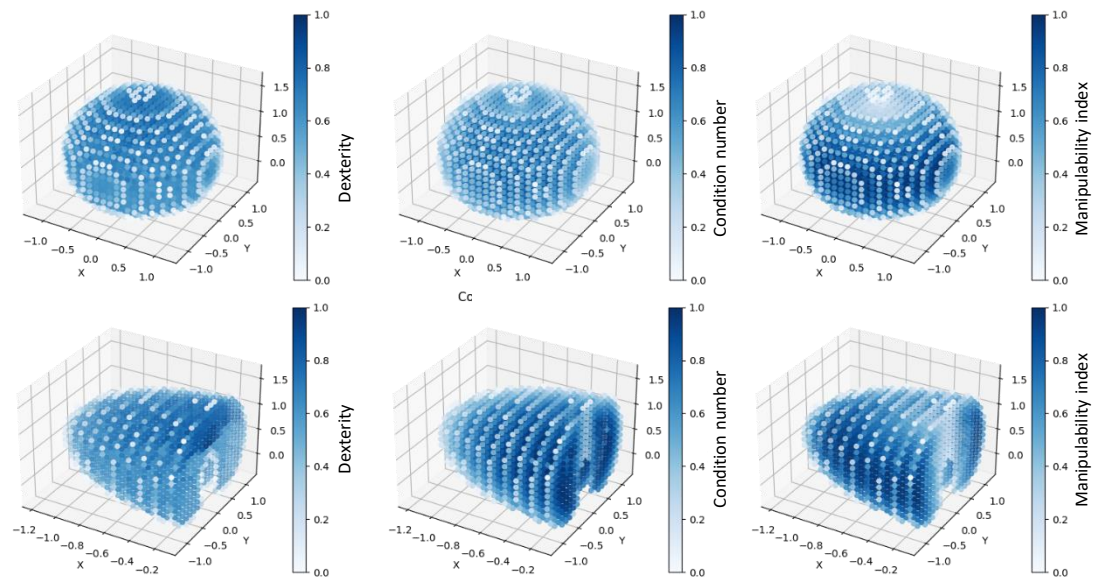
Figure E.1: Part 1: Results of the Design of Experiments of the Performance Maps.



d) Run order 4: 80mm resolution, 25 poses per voxel, robot model ABB IRB 120-0.58

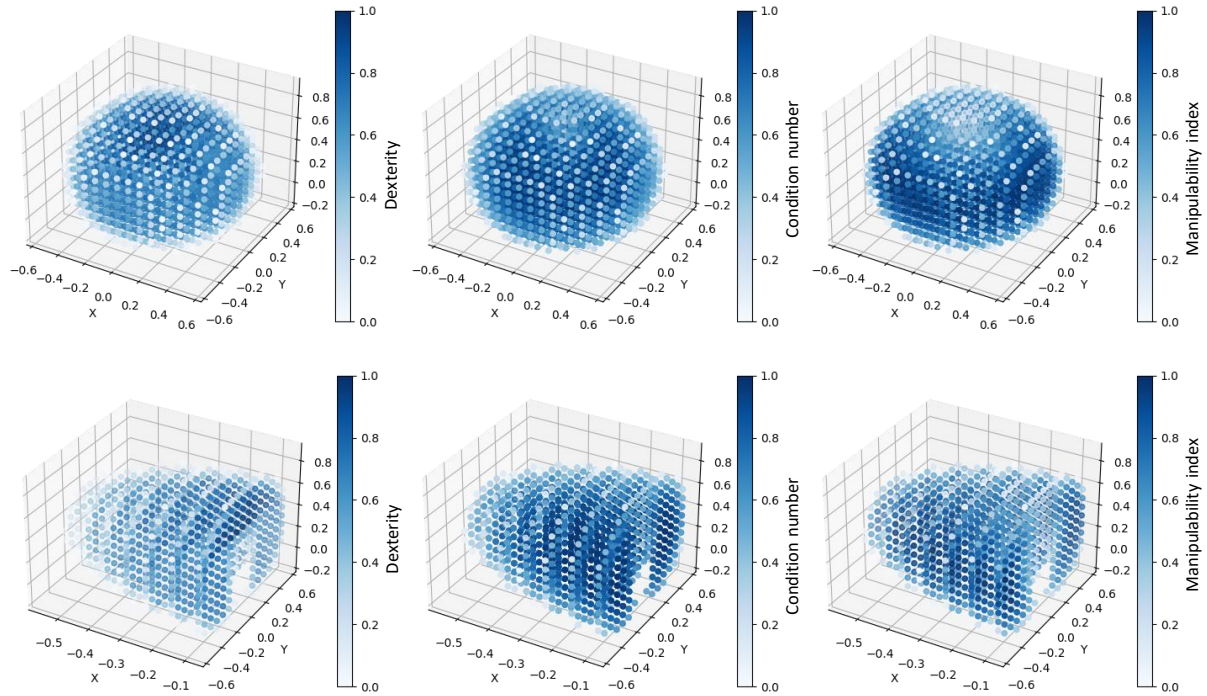


e) Run order 5: 80mm resolution, 50 poses per voxel, robot model ABB IRB 1600-1.2

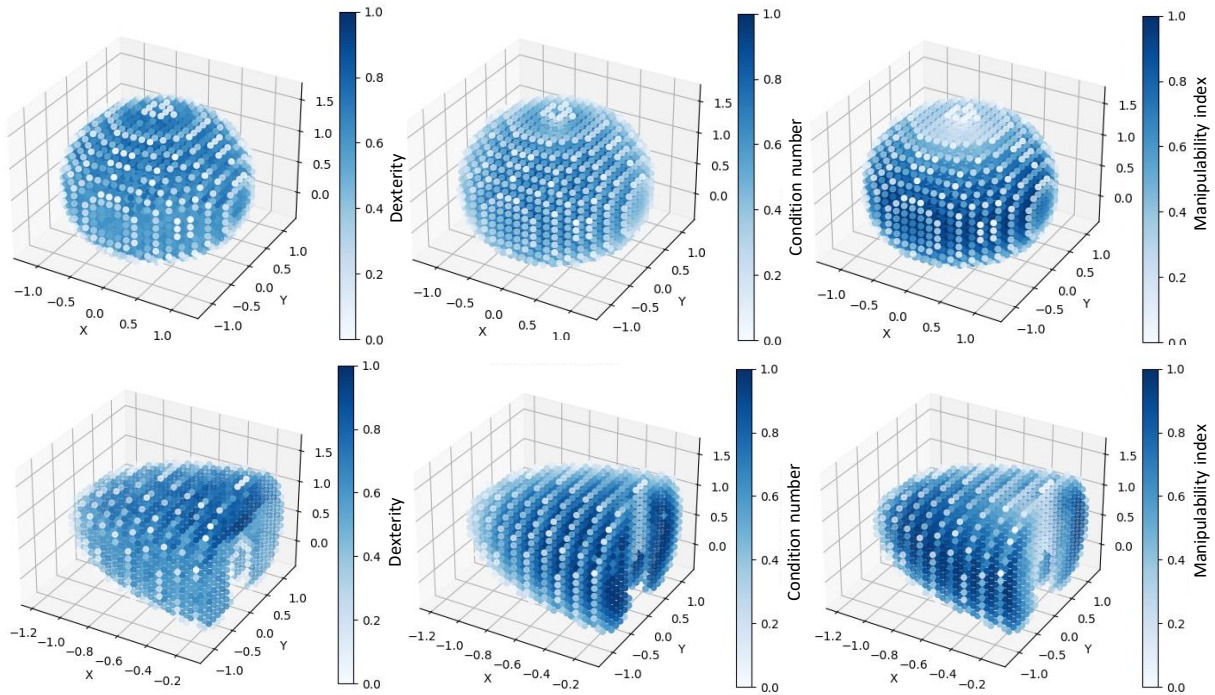


f) Run order 6: 120mm resolution, 50 poses per voxel, robot model ABB IRB 1600-1.2

Figure E.2: Part 2: Results of the Design of Experiments of the Performance Maps.



g) Run order 7: 80mm resolution, 50 poses per voxel, robot model ABB IRB 120-0.58



h) Run order 8: 120mm resolution, 25 poses per voxel, robot model ABB IRB 1600-1.2

Figure E.3: Part 3: Results of the Design of Experiments of the Performance Maps.

F Autonomous clustering-based task allocation and robot base placement

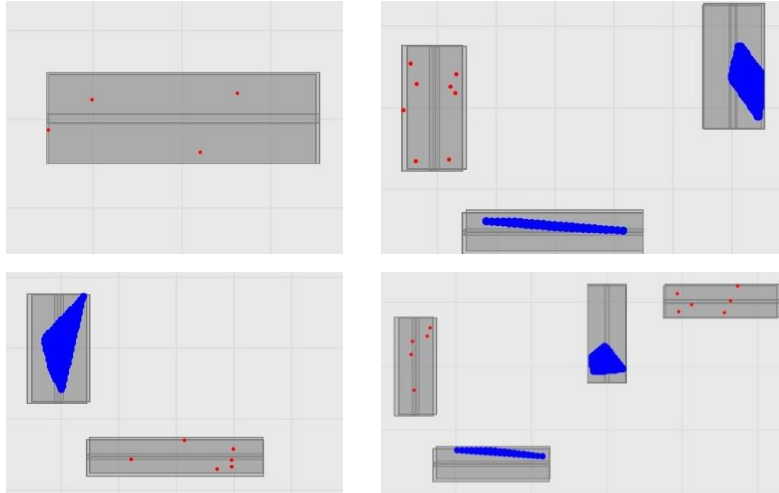


Figure F.1: Partially automated creation of distributed task poses on a random number of products.

In figure F.2, green circles indicate FBPs for the task poses belonging to continuous task sets, whilst the magenta circles belong to the task set containing discrete task poses. The crosses mark the final solution for the base placements for the discrete task poses, whilst the framed and highlighted circles mark the final solution for the base placements of the continuous tasks. The respective colours of the base placements indicate the assignment to the task poses, which are marked in a similar colour.

In figure F.3, blue dots on the left side mark task poses on the product's surface. On the right side, the possible base placements are marked in light green, chosen base placements and allocated task poses are marked in red, blue and yellow.

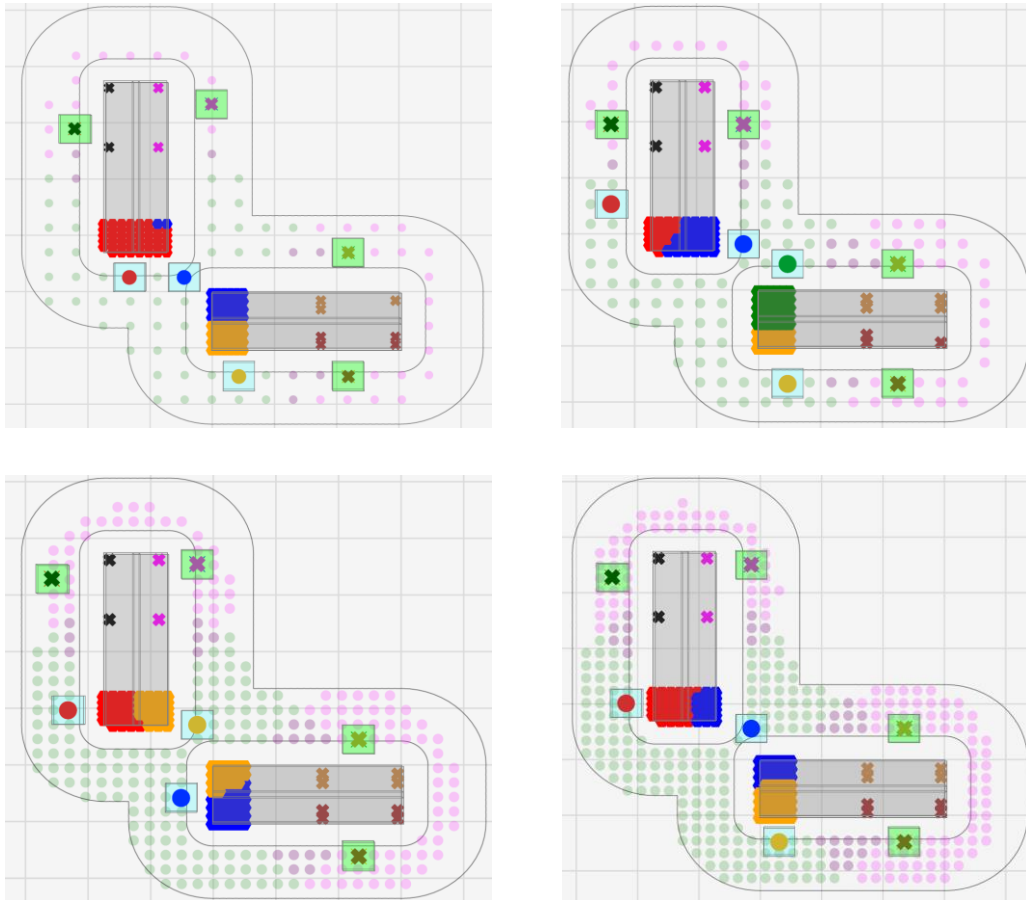


Figure F.2: Test scenarios with increasing search space resolution.

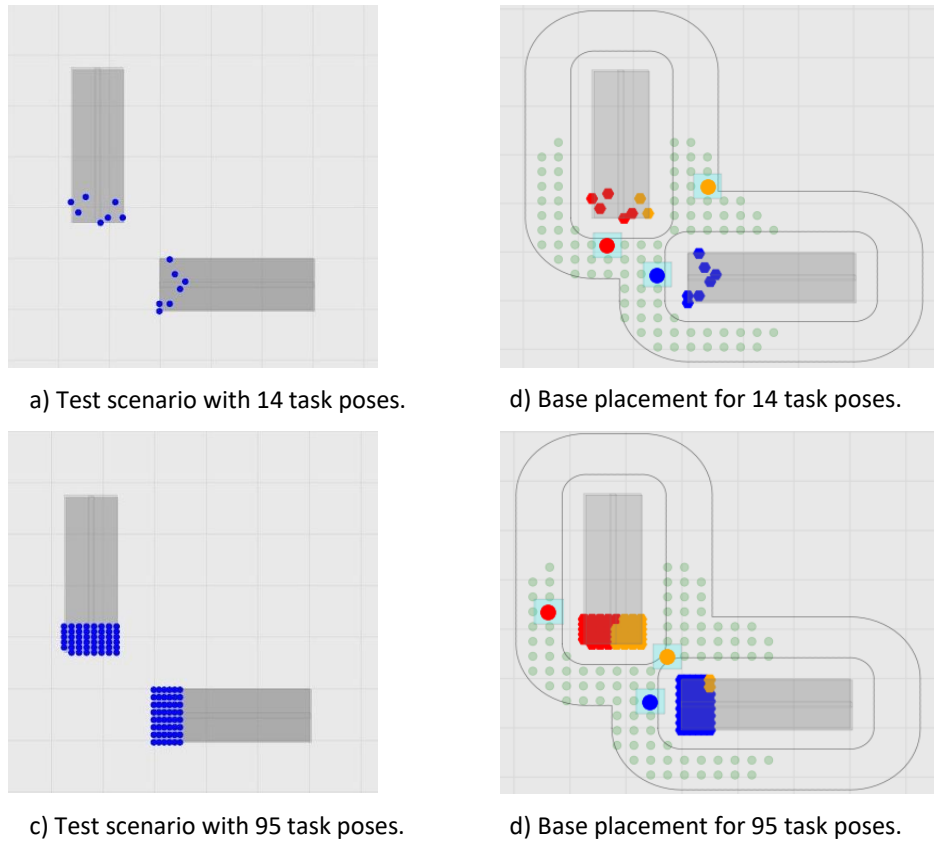


Figure F.3: Test scenarios with constant search space resolution and increasing task poses. Continuation of figure 6.11.

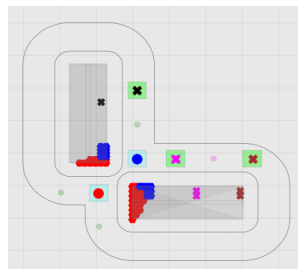


Figure F.4: A test scenario with insufficient resolution fails to identify a solution of robot base placements capable of reaching all task poses.

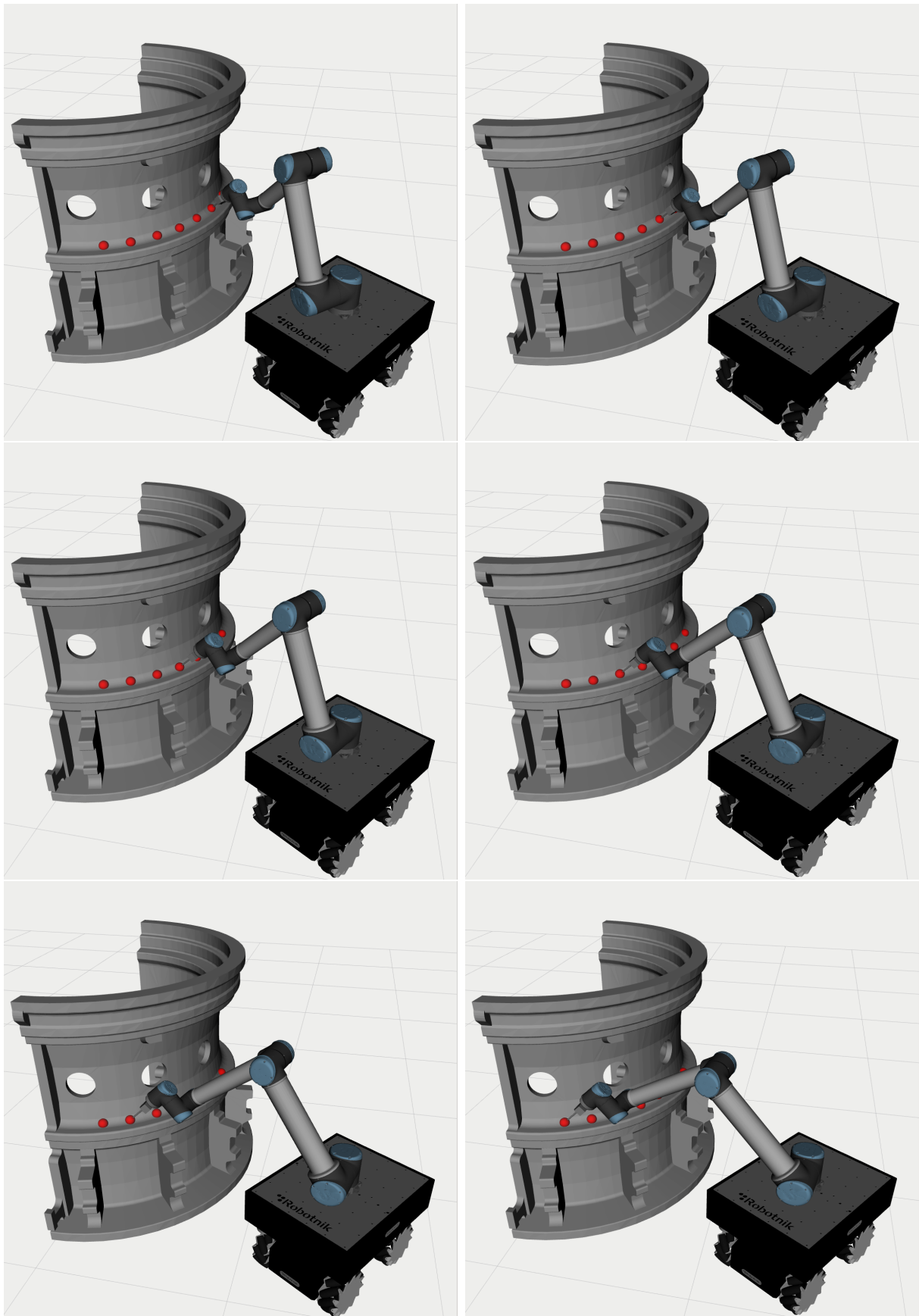


Figure F.5: Validation results showing in simulation the execution of all test poses allocated to the first robot base placement.

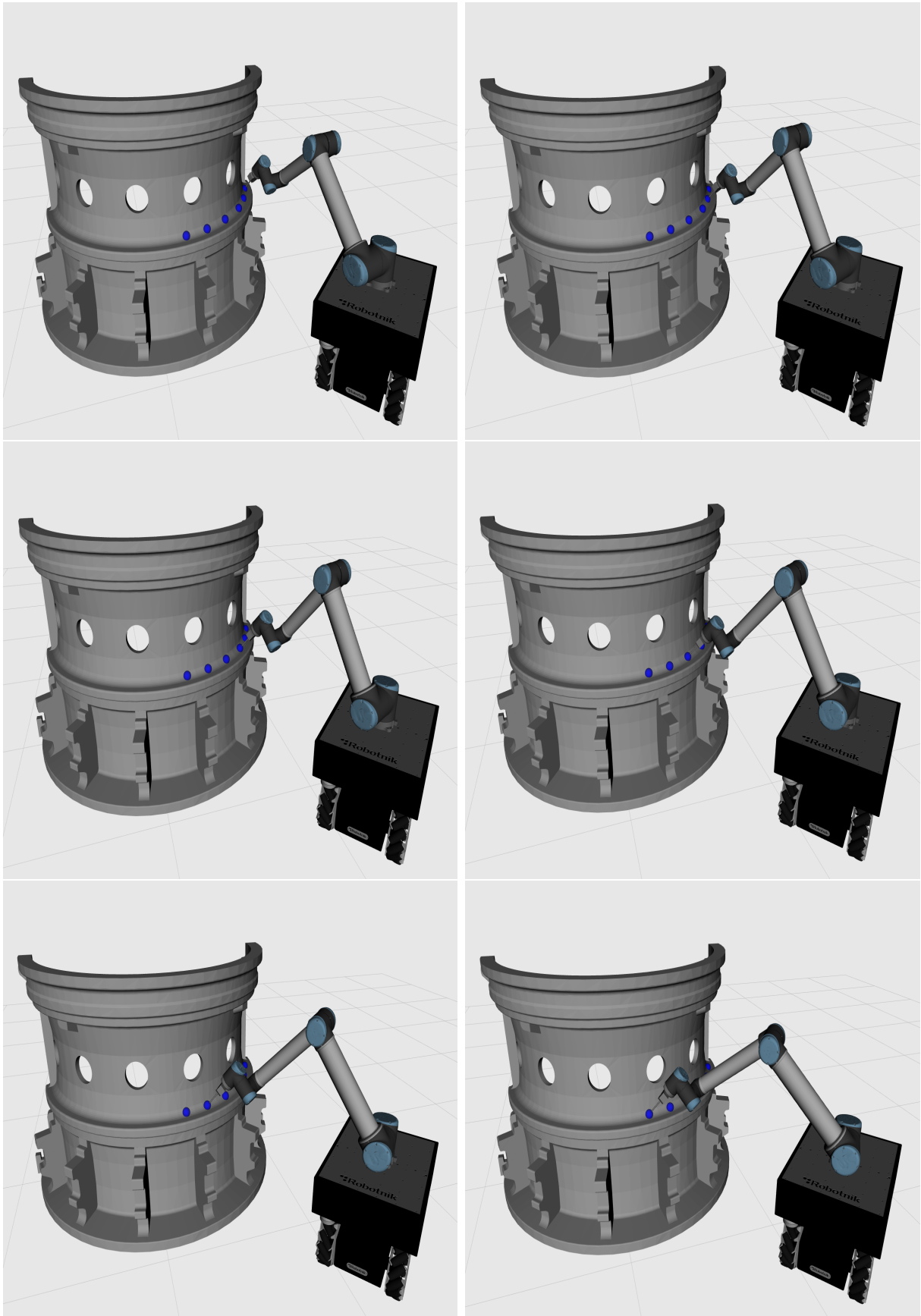


Figure F.6: Validation results showing in simulation the execution of all test poses allocated to the second robot base placement.

G Verification and validation

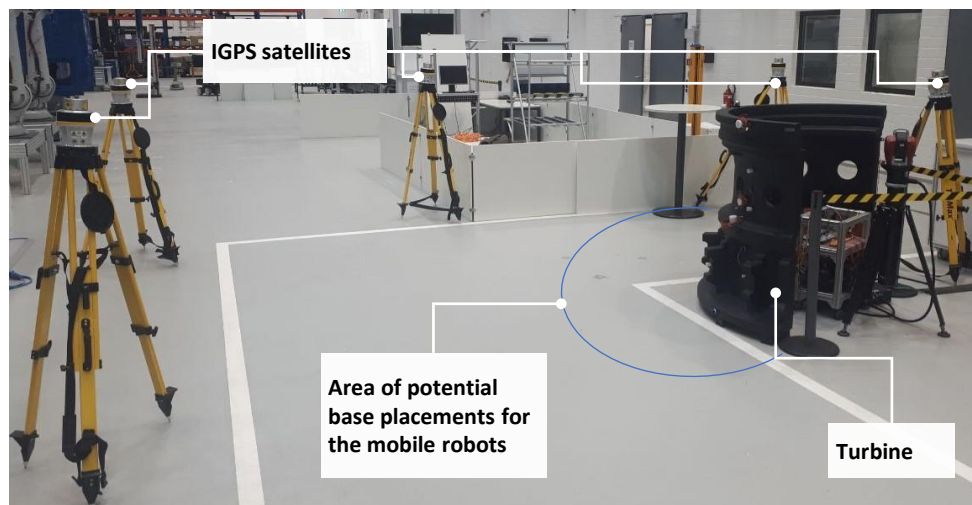


Figure G.1: Metrological setup to span the world coordinate system in the laboratory of the WZL for operational validation of the formation planning.



Figure G.2: Task poses indicated by stickers for marking on the turbine.