

Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen

Samuel Vogel

**INSTITUT FÜR
STATIK UND DYNAMIK
DER LUFT- UND
RAUMFAHRTKONSTRUKTIONEN**
UNIVERSITÄT STUTT GART

Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen

Von der Fakultät für Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von
Dipl.-Phys. Samuel Vogel
aus Bad Waldsee

Hauptberichter: Priv.Doz. Dr.-Ing. Stephan Rudolph
Mitberichter: Prof. Dr. rer. nat. Olaf Deutschmann
Mitberichter: Prof. Dr. rer. nat. habil. Claus-Dieter Munz

Tag der mündlichen Prüfung: 30.11.2015

D93

ISBN 978-3-942807-03-6

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen,
Universität Stuttgart, Stuttgart 2016.

Dieser Bericht kann über das Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Telefon: (0711) 6856-63612, Email: office@isd.uni-stuttgart.de, bezogen werden.

Danksagung

Der größte Dank geht an meine wunderbar(e ,) temperamentvolle Ehefrau Marion. Sie ist trotz meiner recht zeitaufwändigen wissenschaftlichen Aufgaben stets hinter mir gestanden. Durch ihre Stärke und Unterstützung hat sie einen sehr großen Anteil am Gelingen der Arbeit. Lea und ihr bin ich sowohl für das Verständnis dankbar, wenn die Wissenschaft wieder einmal die Oberhand im Familienleben gewonnen hatte, als auch für die zum erfolgreichen Bearbeiten der Arbeit notwendige Ablenkung und Zerstreuung.

Ich danke PD Dr. Stephan Rudolph ganz herzlich für die Betreuung der Arbeit und die vielen geistig anregenden Stunden auf dem Weg zur Universität und am abendlichen Grillfeuer. Stephan hat mir jederzeit die Freiheit gegeben, mich in den mir genehmen Themen entfalten und austoben zu können. Die unzähligen Stunden gemeinsamer wissenschaftlicher Diskussion haben erheblich zum Gedeihen der Arbeit beigetragen. Seiner Frau Diana gebührt ebenfalls großer Dank. Ohne ihr Verständnis für die gelegentlich ausufernden Arbeitszeiten ihres Mannes wäre dieser wahrscheinlich nicht in der Lage, das riesige Engagement und die Gelassenheit aufzubringen, eine so freiheitliche und zugleich fürsorgliche Arbeitsatmosphäre zu schaffen.

Den Professoren Fichter, Deutschmann und Munz sei für die Übernahme des Prüfungsvorsitzes sowie der Mitberichte herzlich gedankt. Bei Professor Markus Till und dem glücklichen Zufall bedanke ich mich dafür, meinen Betreuer Stephan Rudolph überhaupt getroffen zu haben. Der *Arbeitsgruppe Ähnlichkeitsmechanik* danke ich für die Unterstützung und den regen Austausch über Entwurfssprachen und alles was dazu gehört. Besonders bedanken möchte ich mich dabei bei Dr. Johannes Gross, dessen konstruktive Zweifel und Nachfragen erheblich zum inhaltlichen Fortschritt und Reifen der Arbeit beigetragen haben. Peter Arnold von der IILS mbH danke ich für jegliche Unterstützung rund um den Entwurfscompiler und für das zur Verfügung stellen von Literatur.

Meinem Vater danke ich für die Mithilfe in der Korrektur der Arbeit. Der Mutter sei Dank für die Weitergabe naturwissenschaftlichen Interesses. Onkel Josef danke ich für die Nutzung seines Druckers sowie für die Spende der Doktorfeier. Für die Organisation und Durchführung der Feier bedanke ich mich besonders bei Marion, Robère und Bronsen. Anne danke ich für die Korrektur des Abstracts. Nicht unerwähnt bleiben soll die Unterstützung von Joachim Bernhard und Boban in der Prüfungs- und Vortragsvorbereitung. Abschließend möchte ich Bernd Danckert dafür danken, dass er es mir ermöglicht hat diese Arbeit während meiner Tätigkeit in seinem Unternehmen durchzuführen und mich stets ermuntert hat diese Promotion in Angriff zu nehmen. Auch wenn sich die Wege auf Grund geschäftlicher Turbulenzen getrennt haben, gebührt ihm ein großer Anteil am Zustandekommen der Arbeit. Danke!

Ich hoffe die vorliegende Arbeit kann einen kleinen Beitrag dazu leisten, dass die Kombination von Methoden und Arbeitsweisen aus dem Bereich der Physik komplexer Systeme mit graphenbasierten Entwurfssprachen zu einem besseren Verständnis ingenieurwissenschaftlicher Entwurfsprozesse führt. Dem Leser wünsche ich viel Spaß bei der Lektüre und ich hoffe er möge genußvolle und erhellende Momente erleben.

Inhaltsverzeichnis

Danksagung	v
Abkürzungsverzeichnis	ix
Zusammenfassung	x
Abstract	xi
I. Einleitung	1
1. Thematische Einführung	1
1.1. Motivation	1
1.2. Aufgabenstellung	3
1.3. Struktur der Arbeit	4
2. Stand des Wissens	5
2.1. SCR-Systeme	5
2.1.1. SCR Abgasreinigung	5
2.1.2. SCR-System-Entwicklung	7
2.1.3. Entwicklungsmethodik im Bereich SCR	11
2.2. Ingenieurwissenschaftlicher Entwurf	13
2.2.1. Produktentwurf	13
2.2.2. Modellierung	15
2.2.3. Standardisierung	17
2.2.4. Entwurfsbewertung	18
2.2.5. Entwurfsraum	19
2.2.6. Constraint-Satisfaction-Problem	19
2.3. Entwurfssprachen	20
2.3.1. Motivation	20
2.3.2. Entwurfscompiler und Prozessketten	22
2.3.3. Vokabeln (UML-Klassendiagramm)	24
2.3.4. Regeln	26
2.3.5. Produktionssystem (UML-Aktivitätsdiagramm)	26
2.3.6. Lösungspfadgenerator	27
2.4. Computergestützte Pfadsuche und Verrohrung	33
2.4.1. Algorithmen	33
2.4.2. Anwendungen	35
2.5. Komplexe Systeme	36
2.5.1. Nichtlineares System	36
2.5.2. Simulierte Abkühlung	37
2.5.3. Dimensionsbegriffe	38
2.5.4. Ordnungsmechanismen	41

II. Wissensbasierter Entwurf	43
3. Entwurfssprache zur SCR-Systementwicklung	43
3.1. Lösungsweg und Methodik	43
3.1.1. Funktionale Zerlegung	43
3.1.2. Teilentwurfssprache 0: Randbedingungen und Anforderungen einlesen . .	44
3.1.3. Teilentwurfssprache 1: Analytische Auslegung	46
3.1.4. Teilentwurfssprache 2: Packaging der Komponenten	49
3.1.5. Teilentwurfssprache 3: Geometrie der Mischstrecken	52
3.1.6. Teilentwurfssprache 4: Geometrie der Katalysatorboxen	55
3.1.7. Teilentwurfssprache 5: Routing der Rohrstrecken	58
3.1.8. Teilentwurfssprache 6: CFD Berechnung und Bewertung	60
3.1.9. Teilentwurfssprache 7: Thermische Strukturberechnung	64
3.2. Anwendungsbeispiele	66
3.2.1. Baumaschine	66
3.2.2. Frachtschiff	74
4. Verrohrungssprache	79
4.1. Lösungsweg und Methodik	79
4.1.1. Rohrklassen und Regeln	79
4.1.2. Modellrepräsentation der Vokabeln im Starrkörpersimulator	82
4.1.3. Umsetzung der Regeln im Starrkörpersimulator	84
4.1.4. Berücksichtigung weiterer Zwangsbedingungen	93
4.1.5. Produktionssystem der Verrohrungssprache	95
4.2. Anwendungsbeispiele	97
4.2.1. Synthese mit Simulierter Abkühlung	97
4.2.2. Parametrische Optimierung - Reinitialisierung	107
4.2.3. Lösungsmenge der Verrohrung	111
III. Ordnungsmechanismen im wissensbasierten Entwurf	115
5. Selbstassemblierende Lösungspfadfindung	115
5.1. Lösungsweg und Methodik	115
5.1.1. Lösungssequenz aus Abhängigkeiten	115
5.1.2. Lokale Symmetrie	116
5.1.3. Definition eines selbstassemblierenden, lokalen Regelwerkes	118
5.2. Umsetzung und Implementierung	120
5.2.1. Beispiele	120
5.2.2. Äquivalenz zum graphenalgorithmischen Lösungspfadgenerator	120
5.2.3. Komplexitätsbetrachtung	126
5.2.4. Laufzeitenvergleich	128
6. Dimensionsabhängige Entwurfsabfolge	130
6.1. Lösungsweg und Methodik	130
6.1.1. Entwurfsprozess als Nichtlineares System	131
6.1.2. Entwurfsraum und Entwurf als Constraint-Satisfaction-Problem	132
6.1.3. Reihenfolge von Packaging und Routing	135
6.1.4. Dimension der Schnittmenge	138

6.1.5. Dimension und Entwurfsablauf	140
6.2. Begründung und Anwendung	141
6.2.1. Mathematische Begründung der dimensionsabhängigen Entwurfsabfolge .	141
6.2.2. Dimensionsabhängige Entwurfsabfolge und Lösungspfad	144
6.2.3. Anwendung der Dimensionsbetrachtung	145
6.2.4. Einordnung der Ergebnisse	148
6.2.5. Weiterführende Betrachtung: Komplexität und Dimension im Entwurf . .	149
IV. Rückblick	151
7. Diskussion	151
7.1. Inhaltliche Aspekte	151
7.2. Implementatorische Aspekte	152
7.3. Industrieller Einsatz	153
8. Zusammenfassung	155
8.1. Ergebnisse	156
8.2. Ausblick	157
Literatur	162
Lebenslauf	176

Abkürzungsverzeichnis

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CFD	Computational Fluid Dynamics
DIN	Deutsches Institut für Normung
DoE	Design of Experiments
FEM	Finite Elemente Methode
HWL	Harnstoff-Wasser-Lösung
IDE	Integrated Development Environment
ISO	International Organization for Standardization
KBE	Knowledge Based Engineering
KI	Künstliche Intelligenz
KKF	Kegel-Kegel-Fasskreisbogen
NO_x	Stickoxide
PM	Particulate Matter
SCR	Selective Catalytic Reduction
SoS	System-of-Systems
STL	Surface Tessellation Language
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
VLSI	Very-Large-Scale Integration

Zusammenfassung

Die in den Industrie- und Schwellenländern gültige Emissionsgesetzgebung macht bei der Verwendung von Dieselmotoren eine außermotorische Abgasreinigung notwendig. Dies betrifft nicht nur Fahrzeuganwendungen auf der Straße, sondern auch Off-Highwayanwendungen wie Marine, Bahn und Industrie. SCR-Systeme (*selective catalytic reduction systems*) stellen in diesen Anwendungen die bevorzugte Technologie zur Stickoxidreduktion dar. SCR-Systeme sind robust gegenüber Kraftstoffen mit erhöhten Schwefelgehalten und erlauben es, den Basismotor auf höhere Stickoxidrohmissionen zu trimmen. Dies führt im Allgemeinen zu geringeren Kraftstoffverbräuchen und stellt damit ein Beitrag zur CO₂-Reduktion dar. Die individuellen Einsatzprofile und geringen Stückzahlen in den Off-Highwayanwendungen erfordern einen besonders effizienten Entwurfsprozess der SCR-Systeme. Dieser kann mit Hilfe graphenbasierter Entwurfssprachen realisiert werden.

Die wissensbasierte Entwurfsmethode der Entwurfssprachen stellt ein digitales, regelbasiert ausführbares Abbild des Entwurfsprozesses her. Das Entwurfswissen wird in Form von Regeln und Vokabeln abgelegt. Durch automatisierte Analyse- und Simulationsschleifen werden im Produktionssystem Entwurfsentscheidungen auf Basis objektiver Analyseergebnisse getroffen. Ausgehend von gegebenen Anforderungen (Emissionsziele, Motordaten, Bauraum, etc.) werden digitale Entwürfe von SCR-Systemen erzeugt und anhand regelbasiert generierter Simulationsmodelle (z.B. Strömungssimulation) validiert. Damit wird eine Beschleunigung des Entwurfsprozesses von SCR-Systemen um mehr als eine Größenordnung erreicht. Die dafür notwendigen CAD-Geometrien der verwendeten Rohrleitungen werden aus standardisierten Rohrbögen mit konstanten Bogenwinkeln und Radien erzeugt. Für die Synthese dieser Rohrstrecken wird ein Optimierungsalgorithmus (*Simulated Annealing*) eingesetzt. Dabei werden topologische Variationen mit Hilfe einer analytischen Konstruktionsvorschrift (*Kegel-Kegel-Fasskreis*-Konstruktion) dargestellt. Diese erlaubt einen schnellen und robusten Austausch einzelner Bogenelemente unter Beibehaltung der Gültigkeit der Rohrstrecke. Parametrische Änderungen der Rohrstrecke werden mit Hilfe eines Starrkörpersimulators durchgeführt.

Das digitale Abbild des Entwurfsprozesses enthält analytische Gleichungen. Die Lösungssequenz dieser Gleichungen muss bei der Ausführung der Entwurfssprache automatisch bestimmt werden. Dies wird mit Hilfe eines Lösungspfadgenerators realisiert. In der vorliegenden Arbeit wird dazu auf Basis von Symmetriebetrachtungen ein selbstassemblierender Lösungspfadalgorithmus entwickelt. Dieser erlaubt, im Regelfall zyklischer Kopplungen, eine generische Parallelisierung der Lösungspfadfindung und führt so zu einer erheblichen Beschleunigung, verglichen mit klassischen graphenbasierten Algorithmen. Daneben wird ein Mechanismus zur Bestimmung der Abfolge einzelner Entwurfsschritte entwickelt. Durch eine systematische Sequenzialisierung des Entwurfsprozesses kann eine Reduktion der Entwurfskomplexität erreicht werden. Die mit den Produkthanforderungen verträglichen Auslegungen der zu integrierenden Systeme bilden dabei Teilmengen im Raum der Entwurfsfreiheitsgrade. Aus der mathematischen Dimension dieser Teilmengen lässt sich eine bevorzugte Entwurfssequenz bestimmen: Bei der sequenziellen Integration zweier Teilsysteme sollte mit der Auslegung des niederdimensionalen Teilsystems - dem System mit weniger Freiheitsgraden - begonnen werden. Das Prinzip der dimensionsabhängigen Entwurfsabfolge kann neben der Strukturierung des Entwurfsprozesses auch als Basis eines zukünftig selbstorganisierten Entwurfsprozesses dienen.

Abstract

The emission standards in industrialized and emerging countries require the usage of exhaust aftertreatment systems in the application of diesel engines. The emission regulation applies to both on-highway and off-highway applications as navy, rail or industry. SCR (selective catalytic reduction) systems are the preferred technology to reduce nitrogen oxides in these applications. SCR systems are tough against poor fuel qualities with high sulfur levels. The SCR technology allows higher nitrogen oxides raw emissions. In general, this enables engine calibrations with lower fuel consumptions and hence reduced carbon dioxide emissions. Low numbers and individual operation conditions in off-highway applications require an outstanding efficient SCR system design process. Efficient and automated design processes can be realized with graph-based design languages.

Design languages, as a knowledge based engineering method, provide a digital, rule-based and re-executable implementation of the design process. The design knowledge is digitally stored in a design grammar consisting of rules and vocabulary. Automated analysis and simulation loops are used to trigger decisions within the design process. Starting from given product requirements (emission targets, engine data, available installation spaces, etc.) virtual SCR system designs are generated and virtually validated using automatically generated and executed computational fluid dynamic models. Using design languages in SCR system design leads to an acceleration of at least one order of magnitude. The pipe routes are made up from standardized pipe bends with fixed bend angles and radii together with straight pipe elements having arbitrary lengths. Simulated annealing is used for the synthesis of the pipe routes. The geometry based *Kegel-Kegel-Fasskreis* (KKF) algorithm is proposed to conduct topological manipulations of the pipes. Single pipe bends can be quickly replaced using the KKF algorithm without losing the pipe route's validity. Parametric manipulations of the pipes are executed by a rigid body simulation approach.

Solving analytical dimensioning equations is part of the design process. The solution sequence of these coupled dimensioning equations needs to be determined automatically during the execution of the design language. This is carried out by a so-called solution path generator. Within this work a self-assembling solution path generator algorithm is presented. The proposed algorithm is based on symmetry properties of the underlying equation structure. For the common situation of acyclic dependencies a generic parallelization of the self-assembling solution path generation leads to a significant acceleration compared to classical graph-based solution algorithms. Additionally, an order mechanism for the autonomous prediction of the design sequence, composed of iterative design steps, is developed. The complexity of the design process can be significantly reduced by a systematical sequentialization of the design process. Valid subsystem designs that are in accordance with the given requirements define manifolds within the design space. It is shown that the mathematic dimensions of the valid designs' manifolds determine a preferred design sequence: In a sequential integration of two subsystems it should be started with the dimensioning of the subsystem which has the smaller mathematical dimension (less degrees of freedom). A design sequence that is based on the subsystems' mathematical dimensions can either be used to structure the design process or it can be a fundamental concept for a future self-organized design process.

Teil I.

Einleitung

1. Thematische Einführung

In der thematischen Einführung soll zunächst die Motivation für die Verwendung von Entwurfsprachen als wissensbasierte Methodik im Entwurf von Abgasreinigungssystemen für Dieselmotoren gegeben werden. Im weiteren Verlauf wird die Aufgabenstellung der Arbeit herausgestellt und die Struktur der Arbeit vorgestellt.

1.1. Motivation

Abgasreinigungssysteme gehören in den westlichen Märkten zur Standardausrüstung von Personen- und Lastkraftwagen. Eine immer strengere Gesetzgebung zur Luftreinhaltung erzwingt eine Schadstoffreduktion bei Verbrennungskraftmaschinen [58, 92]. Dieselmotoren weisen durch ihre heterogene Gemischbildung (Kraftstoff/ Verbrennungsluftgemisch) im Vergleich zu den im Allgemeinen mit einem homogenen Gemisch betriebenen Ottomotoren¹ prinzipiell höhere Emissionswerte bezüglich Particulate Matter (PM) und Stickoxide (NO_x) auf [127].

Beide Schadstoffkomponenten, Partikel und Stickoxide, stehen in der dieselmotorischen Verbrennung in einem sogenannten Trade-Off: So entstehen Partikel primär in fetten Zonen des inhomogenen Verbrennungsgemisches, während Stickoxide vornehmlich in den mageren Verbrennungszonen mit erhöhter Verbrennungstemperatur gebildet werden. Beide Ursachen stehen damit entgegengesetzt zueinander. Versucht man einen der beiden Entstehungsmechanismen zu unterdrücken, zum Beispiel die Verminderung überfetteter Zonen durch Erhöhung der Turbulenz im Brennraum, so kommt es im Allgemeinen zur stärkeren Ausbildung des jeweils entgegengesetzten Phänomens. Im erwähnten Beispiel führt eine Erhöhung der Turbulenz und eine bessere Durchmischung des Gemisches zu höheren Verbrennungstemperaturen und damit erhöhten Stickoxidemissionen [127].

Auf Grund des höheren Verdichtungsverhältnisses und der im Allgemeinen höheren Verbrennungstemperatur weist der Dieselmotor prinzipbedingt einen höheren Wirkungsgrad als ein Ottomotor auf [127]. Dies macht den Dieselmotor in mobilen Anwendungen abseits der Straße (Off-Highway-Anwendungen), gerade bei höheren Leistungen und hoher Auslastung (Baumaschinen, Bahn, Mining, Schiffe, Stromerzeugung,...), zum bevorzugten Antriebssystem.

Neben der oben dargestellten, vom Gesetzgeber oder multinationalen Organisationen gewollten Verschärfungen der zulässigen Schadstoffemissionsgrenzen, steht im Rahmen der Bekämpfung des Klimawandels die Forderung nach einer gleichzeitigen Reduktion der Treibhausgasemissionen zunehmend im Vordergrund [14]. Für den Verbrennungsmotor bedeutet dies vor allem eine Reduktion des Kohlendioxidaustrittes und damit des Kraftstoffverbrauches [127]. Die schon angedeutete reziproke Beziehung zwischen Partikel- und Stickoxidemissionen steht dabei auch in direktem Zusammenhang mit dem Kraftstoffverbrauch: Hohe Verbrennungstemperaturen, die mit hohen Stickoxidemissionen einhergehen, führen auf einen hohen thermischen Wirkungsgrad

¹Direkteinspritzende Ottomotoren weisen wie Dieselmotoren eine heterogene Gemischbildung auf.

des Dieselmotors [127]. Damit ergibt sich eine positive Korrelation der thermische Effizienz eines Dieselmotors mit dem Stickoxidausstoß.

Im Allgemeinen ist es durch innermotorischen Maßnahmen möglich, zum Beispiel Vorverstellung des Einspritzbeginns, den Betriebspunkt des Motors in Richtung hoher thermischer Wirkungsgrade und damit hoher Stickoxidemissionen und kleiner Verbräuche zu verschieben. Eine innermotorische Verminderung der Stickoxide führt, wie oben angedeutet, prinzipbedingt zu einer Erhöhung sowohl des Verbrauchs als auch der Partikelemissionen. Damit kommt der außermotorischen Reduktion der Stickoxide in der Entwicklung moderner und spritsparender Dieselmotoren und Antriebssysteme eine Schlüsselrolle zu [47, 127].

Aktuell stellt die sogenannte Selektive-Katalytische-Reduktion gerade im Bereich der Off-Highwayanwendungen das Mittel der Wahl [175] zur außermotorischen Stickoxidminderung dar [127, 48]. Dabei wird ein Reduktionsmittel in das Abgas eindosiert, welches auf einem sogenannten Selective Catalytic Reduction (SCR)-Katalysator mit den Stickoxiden des Abgases chemisch reagiert und diese zu weitestgehend unbedenklichen Stoffen umsetzt.

Gerade im Bereich der Off-Highway-Applikationen ist der Entwickler von Abgasreinigungssystemen mit unterschiedlichsten Anforderungen konfrontiert. So reichen die Anforderungen von Anwendungen mit sehr hohen Auslastungsfaktoren (Bagger, kommerzielle Schiffe,...) bis zu Anwendungen mit geringen Auslastungen (Kranantriebe, Yachten,...). Diese Anwendungsvielfalt, sowohl in den Anforderungen als auch in den Randbedingungen der verschiedenen Fahrzeuge (Einbauraum, Betriebstemperaturen, Schwingungsbelastung,...), schlägt sich direkt in der Entwicklung und Applikation von Abgasreinigungssystemen nieder [166].

Diese Spezialisierung der Systeme und ihrer Vielfalt in den verschiedenen Anwendungen [166] führt zu einer Reduktion der Stückzahlen identischer Abgasreinigungssysteme. Dies erfordert eine sehr effiziente Entwicklung maßgeschneiderter Systeme zur Reduktion der Systemkosten bei gleichzeitiger Darstellung optimaler Wirkungsgrade [197].

Solche Abgasreinigungssysteme bedienen sich zur Erhöhung der Anzahl der Gleichteile bevorzugt aus einem Baukastensystem [166]. Aus einem Baukasten von am Prüfstand validierter Komponenten zusammengesetzte Systeme werden idealerweise digital validiert, um das Risiko technischer Schwierigkeiten in der Anwendung so gering wie möglich zu halten. Somit ist ein hocheffizienter virtueller Entwurfsprozess von Abgasreinigungssystemen für die effiziente Entwicklung und Validierung der vielfältigen Off-Highway-Anwendungen unabdingbar.

Dieses Anforderungsprofil erfordert eine Optimierung des Entwurfsprozesses. Neben „klassischen“ Strategien, wie Simultaneous oder Concurrent Engineering, welche auf der Parallelisierung und gleichzeitigen Ausführungen von Entwurfsaufgaben beruhen, kommen mit steigender Rechenleistung immer mehr computergestützte Entwurfsmethoden zum Einsatz [138].

Heute gängige computergestützte Entwurfsmethoden beinhalten meist eine parametrische Integration von Geometrieerstellung (Computer Aided Design (CAD)) und Simulation (Finite Elemente Methode (FEM), Computational Fluid Dynamics (CFD)) und erleichtern so den Datenaustausch zwischen verschiedenen Programmen (Ansys Workbench, Star CCM, Catia,...) [54, 157, 84]. Diese informationstechnische Integration nimmt dem Entwerfer lästige Übersetzungsarbeit ab, stellt aber nur einen Zwischenschritt zu einer vollständigen Automatisierung des Entwurfsprozesses dar, welcher unter Verwendung von Methoden aus dem Bereich der Künstlichen Intelligenz (KI) realisiert werden kann [8, 157, 158].

Graphenbasierte Entwurfssprachen [3, 4] stellen dabei, als eine Verallgemeinerung von aufgaben- und anwendungsspezifischen Entwurfsgrammatiken [8], eine Künstliche Intelligenz (KI) ba-

sierte, computergestützte Entwurfsmethode dar. Entwurfssprachen erlauben in einer abstrakten Beschreibungsform den Entwurfsprozess digital zu erfassen und wiederausführbar zu machen [155, 157]. Diese Automatisierung von Standardaufgaben des Entwurfes verspricht den Entwerfer von stupiden, immer wiederkehrenden Aufgaben zu befreien und so Freiraum für Innovation zu schaffen und gleichzeitig den Entwurfsprozess zu rationalisieren.

Die digitale Abbildung des Entwurfsprozesses im Computer lässt die prinzipielle physikalische Komplexität eines Produktentwurfes unverändert. Dabei bestehen eine Vielzahl von Kopplungen zwischen den Teilsystemen des Produktes beziehungsweise den Bauteilen, die zu funktionierenden Baugruppen kombiniert werden, welche wiederum zu einem Gesamtprodukt zusammengefügt werden sollen. Dieser Gedankengang lässt die hierarchischen (Detail-)Stufen erahnen, welche in der Beschreibung eines Produktentwurfes berücksichtigt werden müssen. Dazu kommen potentielle Kopplungen über verschiedenste Wechselwirkungsmechanismen (mechanisch, geometrisch, physikalisch, logisch,...) zwischen verschiedenen Skalen und Domänen: Mikroskopische Oberflächeneigenschaften können zum Beispiel globale Strömungsfelder beeinflussen oder die mechanische Lebensdauer eines Bauteils bestimmen, welche wiederum, im übergeordneten Kontext einer Baugruppe, für die Gesamtfunktion eines Produktes relevant sein können.

Mit dieser Komplexität und der Vielfalt der Abhängigkeiten muss bei einer computergestützten Automatisierung des Produktentwurfes umgegangen werden. Das Ziel der Automatisierung und Digitalisierung des Entwurfes sollte auch die Verringerung der Test- und Versuchsaufwände sein, welche im klassischen Entwurf typischerweise auftreten. Die vollständige Digitalisierung des Produktentwurfes macht die Erstellung immer besserer und detaillierterer Modelle des Produktes notwendig und steigert so die im digitalen Produktentwurf zu beherrschende Komplexität noch weiter.

Die vorliegende Arbeit soll, den vorhergehenden Gedankengängen folgend, die Automatisierung des Entwurfes von SCR-Abgasreinigungssystemen mit Hilfe von *graphenbasierten Entwurfssprachen* umfassen. Darauf aufbauend sollen mögliche Lösungskonzepte zur Beherrschung der Komplexität des Entwurfes unter Einbeziehung von Ordnungskonzepten, welche unter anderem im Bereich der Analyse komplexer Systeme [178] untersucht werden, erarbeitet werden.

1.2. Aufgabenstellung

Eine große Herausforderung in der automatisierten Generierung eines SCR-Systems stellt die Synthese eines Geometriemodells dar, welches keine Kollision mit den Bauraumgrenzen aufweist. Bei der gleichzeitigen Synthese mehrerer Abgaslinien, für Fahrzeuge mit mehreren Motoren im gleichen Bauraum, sollte es darüber hinaus nicht zu Überschneidungen zwischen den Geometrien der einzelnen Abgaslinien kommen.

Die Komplexität der zu entwerfenden Abgassysteme soll sich dabei auf einsträngige Systeme ohne Strömungsverzweigungen beschränken. Die Rohrstrecken werden unter Verwendung von Standardrohrelementen, Bögen mit festen Biegeradien und Bogenwinkeln, generiert. Diese Aufgabe soll in eine eigene Teilentwurfssprache, die sogenannte *Verrohrungssprache*, ausgelagert und gekapselt werden. Diese Verrohrungssprache soll einen stochastischen Optimierungsalgorithmus, welcher auf Konzepten der statistischen Physik (Stichwort komplexe Systeme) basiert, beinhalten. Damit können Modelle paralleler einsträngiger Abgassysteme in einem gemeinsamen Bauraum synthetisiert werden.

Für eine Bewertung der so erzeugten SCR-Systemgeometrien ist für die Berechnung des Wirkungsgrades die Generierung und Simulation von CFD-Modellen notwendig. Dies beinhaltet

auch die automatisierte Vernetzung (Preprocessing) der erzeugten Geometrie sowie eine Automatisierung der Auswertung (Postprocessing) der Berechnungsergebnisse.

Die in dieser Arbeit erstellte graphenbasierte Entwurfssprache soll, neben den oben genannten Aufgaben der automatisierten Geometrieerstellung und der Generierung des CFD-Modells, die analytische Auslegung der SCR-Katalysatoren umfassen. Neben der Strömungsberechnung sollen auch mechanische Aspekte des Systems, mit Hilfe der Generierung eines Strukturanalysemodells, untersucht werden. Die Erzeugung solcher Modelle kann informationstechnisch analog zur Generierung der CFD-Modelle durchgeführt werden.

Die *Gesamtentwurfssprache* zur automatisierten Ausführung der oben genannten Entwurfsaufgaben soll zur besseren Wiederverwendbarkeit einzelner Module in Teilentwurfssprachen gegliedert sein. Die analytische Vorauslegung, basierend auf der automatisierten Generierung eines entsprechenden Gleichungssystems, wird dann in einer eigenen Teilentwurfssprache ausgeführt. Das generierte Gleichungssystem wird im Entwurfscompiler mit einem *Lösungspfadgenerator* computergestützt verarbeitet und aufgelöst.

Ein Lösungspfadgenerator ordnet ein zu lösendes Zwangsbedingungs- oder Gleichungssystem in die möglichen Lösungssequenzen. Dieser Lösungspfadgenerator soll im Rahmen der Arbeit, unter Verwendung des Ordnungsmechanismus der Selbstassemblierung, weiterentwickelt und zur Verbesserung der Laufzeit parallelisiert werden.

Auf Basis der mit der Entwurfssprache gewonnenen Ergebnisse soll am Ende der Arbeit eine Betrachtung der *mathematischen Dimensionen* der Teilsysteme des Entwurfes erfolgen. Aus dieser Dimensionsanalyse kann ein generischer Ordnungsmechanismus zur Strukturierung des Entwurfsprozesses abgeleitet werden.

1.3. Struktur der Arbeit

Die im vorherigen Abschnitt herausgehobenen Stichwörter stellen die Schwerpunkte der vorliegenden Arbeit dar und finden sich entsprechend in der Gliederung der Arbeit in Abbildung 1.1 wieder. Die Arbeit gliedert sich in vier Hauptteile: Der Einleitungsteil (Teil I), mit thematischer Hinführung (Kapitel 1) und der Darstellung des für die Arbeit als relevant erachteten Stand des Wissens (Kapitel 2), sowie der Schlussteil (Teil IV), mit Diskussion (Kapitel 7) sowie Zusammenfassung und Ausblick (Kapitel 8), umschließen die sich im Titel der Arbeit wiederpiegelnden Hauptschwerpunkte der Arbeit: Den wissensbasierten Entwurf von SCR-Systemen mit Entwurfssprachen (Teil II) und die Identifikation und Anwendung ausgewählter Ordnungsmechanismen zur Strukturierung des Entwurfes (Teil III).

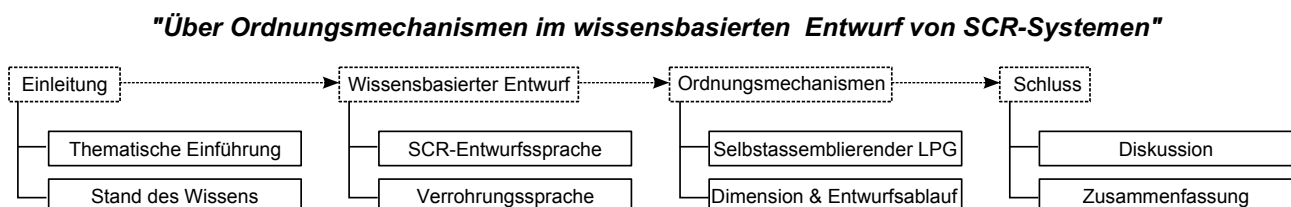


Abbildung 1.1: Hierarchische Aufgliederung der Arbeit in die vier Hauptteile mit jeweils untergeordneten Kapiteln.

Die beiden zentralen Hauptteile sind entsprechend den in der Aufgabenstellung hervorgehobenen Themenschwerpunkten (siehe Abbildung 1.1) weiter aufgliedert:

- SCR-Gesamtentwurfssprache für den SCR-Systementwurf: Automatisierung des Entwurfes von SCR-Systemen (*Kapitel 3*)
- Verrohrungssprache: Synthese von Verrohrungen unter Verwendung von Standardrohrelementen (*Kapitel 4*)
- Selbstassemblierender Lösungspfadgenerator: Weiterentwicklung und Parallelisierung graphenbasierter Algorithmen (*Kapitel 5*)
- Dimension und Entwurfsablauf: Ableitung des Entwurfsablaufes auf Basis einer Dimensionsbetrachtung der beteiligten Teilsysteme (*Kapitel 6*)

Diese Kapitel sind jeweils in die folgenden Untersektionen aufgegliedert: Lösungsweg und Methoden sowie die Darstellung der jeweiligen Ergebnisse anhand von Anwendungsbeispielen. Außerdem sollen diese Kapitel damit thematisch soweit gekapselt sein, dass sie auch isoliert gelesen werden können. Die Betrachtung der mathematischen Dimensionen der am Entwurf beteiligten Teilsysteme von Kapitel 6 baut auf Erkenntnissen auf, die mit den in den vorherigen Kapiteln implementierten und eingesetzten Entwurfssprachen gewonnen wurden.

2. Stand des Wissens

Im vorliegenden Kapitel wird der Stand des Wissens der für diese Arbeit relevanten Themengebiete dargestellt. Zunächst wird auf das in dieser Arbeit zu entwerfenden Objekt, SCR-Systeme, eingegangen. Der Entwurf der SCR-Systeme dient als Beispielanwendung für die entwurfsmethodischen Aspekte der nachfolgenden Kapitel. Es schließt sich eine Einführung in herausgesuchte wissenschaftliche Aspekte der Entwurfsmethodik an. Abschließend werden für diese Arbeit relevante mathematische Grundlagen und Aspekte aus dem Bereich der komplexen Systeme erläutert.

2.1. SCR-Systeme

In diesem Abschnitt soll die Technik und Funktionsweise von SCR-Abgasreinigungssystemen dargestellt werden. Dazu werden zunächst die Funktion von SCR-Systemen erklärt und entwicklungsbezogene Gesichtspunkte dargestellt.

2.1.1. SCR Abgasreinigung

SCR-Systeme dienen zur chemischen Umsetzung schädlicher Stickoxide in den Abgasen von Verbrennungsprozessen in den verschiedensten Anwendungsbereichen. Die Rauchgasentstickung mit Hilfe von SCR-Systemen ist in stationären Anwendungen, vor allem im Kraftwerksbereich, seit mehreren Jahrzehnten als Standardtechnik zur Einhaltung der gesetzlichen Emissionsvorschriften etabliert [81, 153].

Die SCR-Technik wird primär an Verbrennungsmaschinen mit magerer Verbrennung (Sauerstoffüberschuss) eingesetzt und betrifft somit bei den Kolbenmotoren im Wesentlichen die Dieselmotoren. Bei Sauerstoffüberschuss in der Verbrennungskammer bilden sich vermehrt Stickoxide, da der Kraftstoff vollständig verbrennen kann und dieser damit nicht mehr zur chemischen Reduktion entstehender Stickoxide (bei Dieselmotoren im wesentlichen sogenanntes thermisches NO_x) beiträgt.

Neben den Stickoxiden entstehen in Verbrennungsprozessen, über die üblichen Verbrennungsprodukte H_2O und CO_2 hinaus, unverbrannte Kohlenwasserstoffe, Kohlenmonoxid, Ruß und Partikel als unerwünschte und als schädlich deklarierte Emissionen [143, 127]. Diese Emissionen unterliegen in den stationären und mobilen Anwendungen einer immer strenger werdenden gesetzlichen Reglementierung (beispielsweise Abgasgesetzgebung in der Europäischen Union [58]).

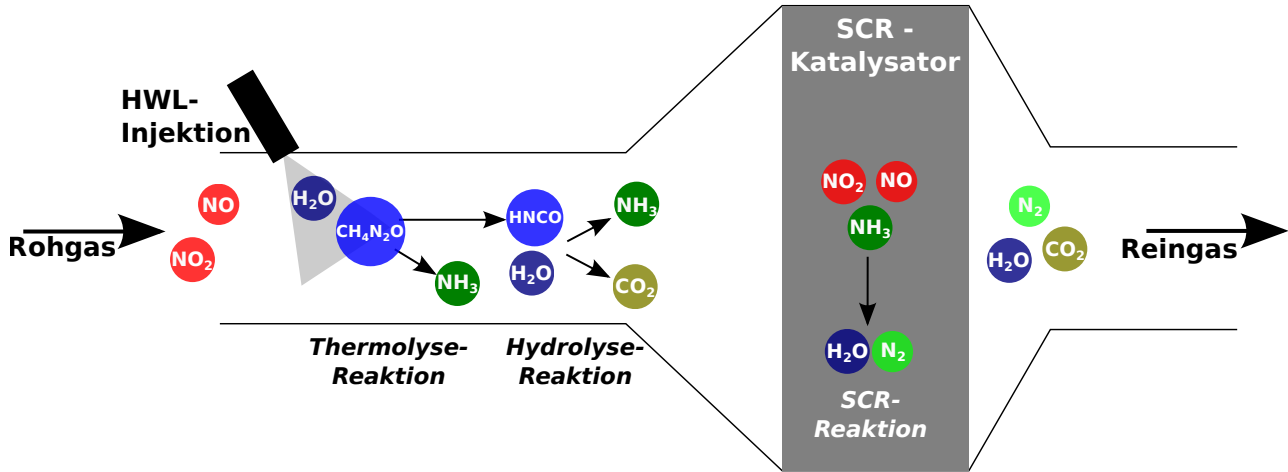
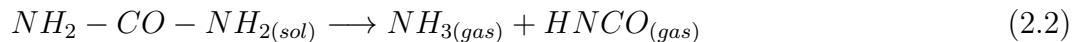
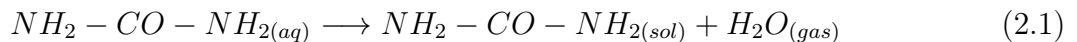


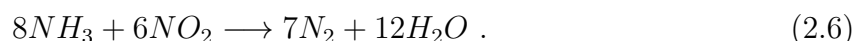
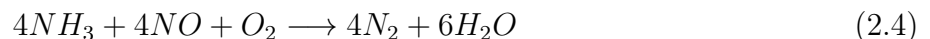
Abbildung 2.1: Schematisches SCR-System und chemische Reaktionen mit Harnstoff-Wasserlösung (HWL) als Reduktionsmittelvorläufer.

Bei der Stickoxidreduktion von Abgasen mit dem SCR-Verfahren wird dem Abgas ein Reduktionsmittel (Ammoniak) oder ein Reduktionsmittelvorläufer (HWL) beigemischt (Abbildung 2.1). Als Reduktionsmittelvorläufer im mobilen Bereich hat sich wässrige Harnstofflösung ($CH_4N_2O_{(aq)}$) durchgesetzt [105, 97], welche im heißen Dieselabgas in einer zweistufigen Thermolysereaktion (2.1) und Hydrolysereaktion (2.2) das Reduktionsmittel Ammoniak (NH_3) sowie Nebenprodukte freisetzt [141]:



Diese Hydrolyse-/Thermolysereaktionen laufen in der Abgasstrecke stromabwärts der Eindosierung ab und können katalytisch beschleunigt werden. Dazu werden Katalysatorträger (z.B. Metall- oder Keramiksубstrate mit Kanal- oder Wabenstrukturen) mit einer katalytisch wirksamen Hydrolysebeschichtung versehen. Diese beschleunigt die chemische Ammoniak-Freisetzung und führt zu einer verbesserten Tropfenverdampfung [142] womit sich die zur Ammoniak-Freisetzung notwendige Abgasstrecke sowie das nachgeschaltete SCR Katalysatorvolumen bei gleichem Wirkungsgrad reduzieren lässt [83].

In der eigentlichen SCR-Reaktion werden die Stickoxide im Abgas mit dem in der Thermolyse/Hydrolyse freigesetzten Ammoniak umgesetzt. Der SCR-Prozess wird mit folgenden Reaktionen beschrieben, die in Abhängigkeit des vorliegenden NO_2 zu NO Verhältnisses ablaufen [64]:



Diese Reaktionen müssen katalysiert werden, da die Reaktionsgeschwindigkeiten der Stickoxidumsetzung sonst zu niedrig wären. Die Reaktion (2.4) wird als Standard-SCR-Reaktion bezeichnet, da sie bei den für Dieselmotoren typischen niedrigen NO_2 zu NO Verhältnissen (in der Regel kleiner 10%) bevorzugt abläuft. Wird dem SCR-System (Eindosierung und SCR-Katalysator) stromaufwärts ein Oxidationskatalysator vorgeschaltet, so wird ein Teil des Stickstoffmonoxides mit dem im Abgas vorhandenen Restsauerstoff zu Stickstoffdioxid NO_2 umgesetzt. Ein paritätisches Verhältniss von NO_2 zu NO stellt die ideale Voraussetzung für die so genannte "schnelle" SCR-Reaktion (2.5) dar [113]. Diese läuft am SCR Katalysator wesentlich schneller ab als die beiden anderen SCR-Reaktionen.

Auf Grund der Temperaturabhängigkeit chemischer Reaktionen wird durch das Vorschalten eines Oxidationskatalysators vor allem die Niedertemperaturaktivität (Temperaturen zwischen 180°C und 300°C) des SCR-Systems durch die gezielte Erzeugung von NO_2 gesteigert. In diesem Temperaturfenster ist die Geschwindigkeit der Stickoxidumsetzung vor allem reaktionskinetisch bestimmt [107]. Bei höheren Temperaturen hängt die Reaktionsgeschwindigkeit im Katalysator von thermodynamischen Gegebenheiten ab [113]. In diesem Regime kommt dem Stoffaustausch (Diffusion, Konvektion) an der katalytischen Oberfläche die geschwindigkeitsbestimmende Rolle zu [98], so dass der NO_2 -Anteil im Abgas für die Reaktionsgeschwindigkeit unerheblich wird.

Der Anteil des am vorgeschalteten Oxidationskatalysator umgesetzten Stickstoffmonoxides hängt von der Auslegung des Oxidationskatalysators als auch vom Betriebspunkt des Motors (vor allem Abgasmassenstrom und Temperatur) ab, so dass nicht in allen Betriebspunkten die "schnelle" SCR Reaktion sichergestellt werden kann.

Für die katalytische Beschichtungen der SCR-Katalysatoren gibt es verschiedene chemische Formulierungen, vor allem abhängig vom Abgastemperaturbereich der jeweiligen Anwendungen. Im Wesentlichen stehen Vanadium-Pentoxid-Beschichtungen und Cu- oder Fe-Zeolit Beschichtungen zur Verfügung. Die Vanadium-Pentoxid-Beschichtung ist kostengünstig und hat sehr gute Wirkungsgrade in einem für viele Anwendungen typischen mittleren Abgastemperaturfenster ($300^\circ\text{C} - 400^\circ\text{C}$)[106]. Ein vorgeschalteter Oxidationskatalysator ist damit nicht notwendig. Desweiteren ist die Vanadium-basierte Beschichtung robust gegenüber hohen Schwefelkonzentrationen im Kraftstoff und Abgas [69]. Diese Katalysatoren werden aber durch hohe Temperaturen ($>500^\circ\text{C}$) zerstört. Dagegen weisen Zeolit-basierte Katalysatoren ein wesentlich robusteres Hochtemperaturverhalten auf. Jedoch ist ein vorgeschalteter Oxidationskatalysator zur NO_2 Erzeugung notwendig, um ähnlich hohe Umsatzraten wie mit Vanadium-basierten Systemen zu erhalten [51].

2.1.2. SCR-System-Entwicklung

In diesem Abschnitt sollen die in der Entwicklung und Auslegung von mobilen SCR-Systemen wichtigsten Komponenten und Größen kurz dargestellt werden. Als wichtigste physikalische Eingangsgrößen des Motors für die Auslegung eines SCR-Systems stellen sich der Abgasmassenstrom, die Abgastemperatur und die Konzentrationen oder Massenströme der zu reduzierenden Emissionen dar.

Druckverlust In der Entwicklung von Abgassystemen stellt der Strömungswiderstand (entspricht dem Druckverlust) des Abgassystems eine sehr wichtige Bewertungsgröße dar. Dieser muss möglichst niedrig gehalten werden, da hohe Strömungswiderstände u.a. zu hohen Verbräu-

chen des Verbrennungsmotors führen und bei Extremwerten sogar zu einem Ausfall des Motors führen können (Behinderung des Gaswechsels / Reduktion der Turboladerfunktion) [127].

Für die Berechnung des Druckverlustes von Standardkomponenten gibt es umfangreiche Sammlungen analytischer Ausdrücke, z.B. in [199]. Deren Einsatz bietet sich gerade im Vorentwurf von SCR-Systemen im Rahmen analytischer Vorauslegungen an [197].

Die Strömungswiderstandskoeffizienten α und β der Katalysatorträger sind von den jeweiligen Herstellern tabelliert erhältlich. Der Druckverlust Δp eines Katalysatorträgers der Länge L lässt sich in guter Näherung mit Hilfe der Darcy-Forchheimer Gleichung für poröse Medien bestimmen [25] (dynamischen Viskosität ν , mittlere Gasgeschwindigkeit v , Dichte ρ):

$$\frac{\Delta p}{L} = \alpha \cdot \nu \cdot v + \beta \cdot \rho \cdot v^2 . \quad (2.7)$$

Für die analytische Bewertung des Druckverlustes der weiteren Anlagenbauteile, die im Allgemeinen turbulent durchströmt werden, bietet sich der von der Reynoldszahl Re abhängige Druckverlustbeiwert $\zeta = \zeta(Re)$ für turbulente Strömungen an:

$$\Delta p = \frac{1}{2} \cdot \zeta \cdot \rho \cdot v^2 . \quad (2.8)$$

Dosiersystem Das Dosiersystem dient zur Einbringung der Harnstofflösung im Abgasstrang. Am Markt gibt es grundsätzlich zwei verschiedene Dosierprinzipien: Neben der luftunterstützten Zerstäubung der HWL-Lösung (air assisted) mit Zweistoff-Düsen steht die ohne zusätzliche Druckluft arbeitende (airless) Sprayerzeugung mit einer Einstoff-Druckdüse zur Verfügung. Der Beitrag [89] enthält eine tiefergehende Gegenüberstellung der Spraycharakteristik von luftunterstützten und luftlosen HWL-Dosiersystemen. Daneben sind Urea-Injektoren mit verschiedenen geometrischen Sprayformen (z.B. Flachstrahl, Konus, Hohlkegel) erhältlich.

Das Dosiersystem besteht aus einem Pumpenmodul, in das im Allgemeinen die Steuerelektronik direkt integriert ist. Ergänzt wird es durch das Injektormodul, welches direkt am Abgasstrang montiert wird und für die Dosierung und Zerstäubung der HWL in den Abgasstrang sorgt. Für die Funktionalität des SCR-Systems ist die Spraycharakteristik des HWL-Injektors entscheidend, da diese über ihren Einfluss auf den Tropfenstransport und die Tropfenverteilung im Abgas maßgeblich die wirkungsgradbestimmende Reduktionsmittelverteilung (siehe Kapitel 2.1.2) am SCR-Katalysator beeinflusst.

Im Dosiersystem wird die Dosiermenge eingestellt. Diese wird als stöchiometrisches Dosierverhältnis α der Konzentration von Ammoniak bezogen auf die NO_x -Konzentration im Abgas, jeweils vor dem SCR-Katalysator, angegeben (2.9) [202]. Zur Bestimmung der HWL-Dosiermenge bei gegebenen α und gegebenem NO_x -Massenstrom wird im Dosiersystem stöchiometrisch das HWL-Äquivalent der benötigten Ammoniakmenge bestimmt:

$$\alpha = \frac{c_{\text{NH}_3}}{c_{\text{NO}_x}} . \quad (2.9)$$

Ein Dosierverhältnis von $\alpha = 1.0$ entspricht gerade der zur vollständigen Umsetzung der Stickoxide nach den chemischen Reaktionssystemen (2.4)-(2.6) stöchiometrisch benötigten Ammoniak- beziehungsweise HWL-Menge. NO_x wird bei der Berechnung der Emissionen im Allgemeinen als NO_2 behandelt.

Hydrolysestrecke Die Abgasstrecke zwischen Eindosierung der HWL und dem SCR-Katalysator wird als Hydrolysestrecke bezeichnet. Innerhalb der Hydrolysestrecke muss eine möglichst gleichmäßige Durchmischung des Abgases mit dem Reduktionsmittel beziehungsweise seinen Vorläufersubstanzen sichergestellt werden, so dass der SCR-Katalysator gleichmäßig beaufschlagt wird und so das Katalysatorvolumen voll ausgenutzt wird [102, 30]. Die Auslegung der Hydrolysestrecke hat damit einen sehr großen Einfluss auf die Effizienz des gesamten SCR-Systems und stellt somit eine Schlüsselkomponente in der Entwicklung von SCR-Systemen dar [74]. Durch die variable Gestaltung der Hydrolysestrecke sowohl die Strömungsführung betreffend als auch durch die Integration optionaler Zusatzkomponenten wie Mischer [211, 150, 23] oder Hydrolysekatalysatoren [142, 83, 48, 166] kann das SCR-System hin zu hohen Effizientwerten optimiert werden.

Die in der Hydrolysestrecke eingesetzten Mischer dienen primär zur radialen Mischung und Vergleichmäßigung des Reduktionsmittels über den Querschnitt des Abgasstromes [151]. Dies geschieht über die Erzeugung von Turbulenz in der Strömung. Die Mischer unterstützen desweiteren die Verdampfung der HWL-Tropfen durch die Erhöhung der Relativgeschwindigkeit von Tropfen und Gas, [176] sowie durch den Tropfenzerfall beim Aufprall und der Wechselwirkung mit den Mischerstrukturen [121, 13].

Die in der Hydrolysestrecke einsetzbaren Hydrolysekatalysatoren sind mit einer Hydrolysebeschichtung aus TiO_2 versehen und beschleunigen so katalytisch die Reaktionen (2.1) und (2.2) der Freisetzung von Ammoniak aus Harnstofflösung [80]. Die Hydrolysekatalysatoren können darüber hinaus auch noch turbulenz erzeugende, radial durchlässige Strukturen beinhalten, welche einen radialen Konzentrationsausgleich und damit eine Querverteilung des Reduktionsmittels im Hydrolysekatalysator bewirken. Strukturierte Hydrolysekatalysatorträger können durch ihre geometrische Struktur die Verdampfung der Tropfen zusätzlich verbessern [85].

Wie beschrieben bestimmen vor allem die Zusatzkomponenten Mischer und/oder Hydrolysekatalysator, im Zusammenspiel mit der strömungsführenden Geometrie der Hydrolysestrecke, die Verteilung sowohl der gasförmigen als auch der flüssigen Anteile des Reduktionsmittels über dem Strömungsquerschnitt. Die Verteilung der Komponenten über den Strömungsquerschnitt wird mit dem sogenannten Uniformity Index ($0 \leq U_i \leq 1$) beschrieben. Der U_i -Wert gibt den Flächenanteil des betrachteten Querschnittes wieder, der von der Komponente effektiv ausgefüllt wird. Als Faustformel gelten U_i -Werte größer 0.95 (Katalysatorvolumen wird zu über 95% ausgenutzt) als gute Verteilungswerte. Der U_i eines Skalares x berechnet sich nach folgender Formel [20] (x sind ortsabhängige Werte des Skalares über Querschnitt Q ; \bar{x} ist über Querschnitt Q gemittelter Wert des Skalares):

$$U_i = 1 - \frac{1}{2} \cdot \frac{\int_Q |\bar{x} - x| dQ}{\bar{x}} . \quad (2.10)$$

Der Uniformity-Index kann dabei für die dimensionslose Bewertung von Strömungskomponenten an einem Querschnitt im Abgasstrang verwendet werden (Abgas, tropfenförmiges beziehungsweise gasförmiges Reduktionsmittel, Ammoniakverteilung...). Ein Mischer zur Verbesserung der Querverteilung des Reduktionsmittels kann beispielsweise über den Vergleich der Uniformity Indizes stromaufwärts und stromabwärts des Mixers beurteilt und bewertet werden. Eine weitere wichtige Charakterisierung der Mischer- oder Hydrolysekatalysatorelemente stellt der Druckverlust des jeweiligen Elementes innerhalb des Abgassystems dar. Dieser wird im Falle von individuellen Mischerelementen in der Regel mit CFD bestimmt. Für Hydrolysekatalysatorelemente stehen aber auch tabellierte Druckverlustwerte der Hersteller zur Verfügung.

Ablagerungen Beim Entwurf der Hydrolysestrecke gilt es neben der Verteilung des Reduktionsmittels ein weiteres Phänomen zu beachten: Der Harnstoff und seine Zersetzungsprodukte können auf den von Abgas durch- oder umströmten Strukturen kristalline Ablagerungsprodukte bilden, welche im Betrieb des SCR-Systems zu einer Verblockung des Abgassystems führen können. Diese Verblockungen können bis zu einem Motorausfall führen [184, 166], da die Ablagerungen in der Regel nicht wieder aufgelöst werden (Abgastemperaturen zu niedrig für eine Zersetzung der Ablagerungen), was zum Beispiel in Marineanwendungen bei Verlust der Antriebskraft zu Manövrierunfähigkeit führen kann. Die kristallinen Ablagerungsprodukte entstehen bevorzugt an Innenflächen der Hydrolysestrecke des Abgassystems durch den Wandkontakt von HW-Lösung aus Nebenreaktionen der Harnstoffzersetzungsprodukte [170]. Durch eine lokale Temperaturabsenkung der Wand des Abgassystems, in Folge des Auftreffens von flüssigen HWL-Tropfen, können über die Zwischenstufe einer Wandfilmbildung solch unerwünschte Ablagerungen entstehen [20, 22].

Der Vorgang der Ablagerungsbildung kann mit den am Markt verfügbaren Strömungssimulationsprogrammen (noch) nicht aufgelöst werden, da bei der Ablagerungsbildung neben der Wandtemperatur auch mikroskopische Oberflächeneffekte eine Rolle spielen. So kommt es beispielsweise an Schweißnähten bevorzugt zur Bildung von Ablagerungen. Die Verhinderung der Ausbildung von Ablagerungen stellt eine notwendige Bedingung eines zielführenden SCR-Systementwurfes dar.

Zur Verhinderung der Ablagerungsbildung muss im Entwurf der SCR-Systeme angestrebt werden, den Wandkontakt des Reduktionsmittels zu verhindern. Die Wahrscheinlichkeit von Wandablagerungen kann mit Hilfe von mehrphasigen CFD-Berechnungen abgeschätzt werden. Dabei kann die Ausbreitung von Tropfen im Strömungsfeld als Lagrange-Phase innerhalb einer CFD-Simulation vorausberechnet werden [61].

SCR Katalysator Die Hydrolysestrecke wird durch den SCR-Katalysator abgeschlossen. Am SCR-Katalysator findet die Reduktion der Stickoxide entsprechend den Gleichungen (2.4)-(2.6) statt. Der SCR-Katalysator wird im Wesentlichen über seine NO_x -Konversionsleistung charakterisiert (siehe z.B. [200, 113]):

$$U_{\text{NO}_x} = \frac{c_{\text{NO}_x \text{ Ein}} - c_{\text{NO}_x \text{ Aus}}}{c_{\text{NO}_x \text{ Ein}}} . \quad (2.11)$$

Die wichtigste Auslegungsgröße zur konzeptionellen Vorauslegung des Katalysatorvolumens V_{Kat} ist die Raumgeschwindigkeit RG . Diese berechnet sich aus dem Normvolumenstrom \dot{V}_{norm} des Abgases zu [56]:

$$RG = \frac{\dot{V}_{\text{norm}}}{V_{\text{Kat}}} . \quad (2.12)$$

Die inverse Raumgeschwindigkeit ist dabei ein Maß für die Aufenthaltsdauer des Abgases im Katalysator. Für eine erste Auslegung wird hier typischerweise auf Erfahrungswerte zurückgegriffen. Anlagen mit niedrigen Anforderungen bezüglich Lebensdauer, Reduktion und qualitativ hochwertigen Kraftstoffen weisen Raumgeschwindigkeiten von ca. 100000 1/h auf (z.B. Yachtanwendungen). Für Anwendungen mit extremen Lebensdaueranforderungen und minderwertigen Kraftstoffen (z.B. im gewerblichen Marinebereich) sind Werte bis 10000 1/h anzutreffen.

Im weiteren Detail hängt die Konversion eines SCR-Katalysators auch von der Abgastemperatur (Abbildung 2.2), der Gaszusammensetzung (vor allem NO_2/NO_x -Verhältnis) und der

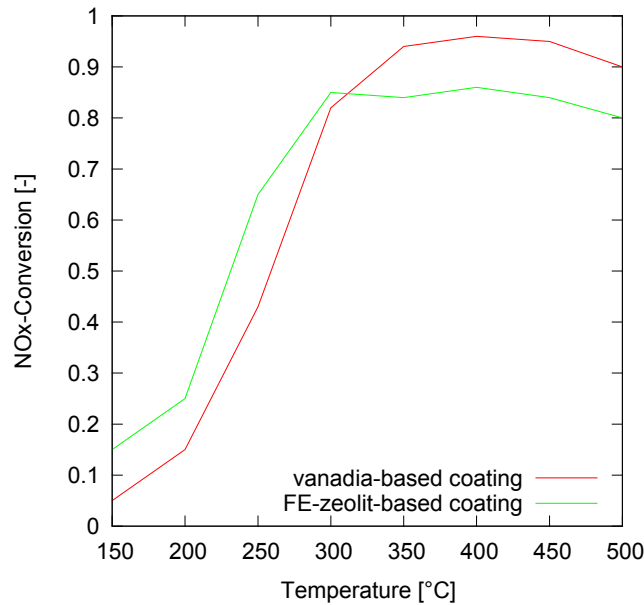


Abbildung 2.2: Typische SCR-Umsatzkurven verschiedener SCR-Beschichtungen bei identischen (Strömungs-)Bedingungen (nach [16]).

stöchiometrischen Dosiermenge α ab. Die Art der Beschichtung, die Beschichtungsmenge sowie die Zelldichte des Trägers sind weitere die Konversion bestimmende Parameter [16].

Die dargestellte Konversionskurve gilt für optimal beaufschlagte Katalysatoren. Eine ungleichförmige Anströmung der Katalysatorträger bezüglich des Reduktionsmittels und der Stickoxide im Abgas führt zu einer Verschlechterung der Konversion. Die typischen Kanalstrukturen der Katalysatorträger erlauben im Allgemeinen keinen radialen Gasaustausch im Katalysator. Damit bestimmt die (inhomogene) Verteilung des Reduktionsmittels am Katalysatoreintritt die Verteilung des Reduktionsmittels über die ganze Katalysatorlänge. Auch Katalysatoren mit radial durchlässigen Strukturen können diese Inhomogenitäten nur zu einem gewissen Grad ausgleichen [27]. Damit ergibt sich, bei inhomogener Anströmung und inhomogener Verteilung des Reduktionsmittels, eine zwischen den Katalysatorkanälen variable und von dem mittleren Dosierverhältnis α abweichende, lokale stöchiometrische Dosiermenge α_{lok} .

Dies führt gerade bei hohen Dosiermengen ($\alpha \rightarrow 1$) zu einer Reduktion der tatsächlichen Konversionsleistung, verglichen mit der maximal möglichen Konversion nach Abbildung 2.2, durch die inhomogene Verteilung der Reaktanden am Katalysatoreintritt. Die Kanäle mit überstöchiometrischen Dosiermengen $\alpha_{lok} > 1$ führen zu einem Ammoniakdurchbruch in dem jeweiligen Kanal, welcher sich als Ammoniakdurchbruch $c_{NH_3,Aus}$ im Abgas stromab des SCR-Katalysators darstellt. Diese Ammoniakemission stellt eine weitere wichtige Auslegungsgröße dar [204], da sie in der Emissionslegislativ eine reglementierte Größe darstellt [58]. Die Verteilungen des Reduktionsmittels und des Abgases am Eintritt des SCR-Katalysators spielen damit in der Auslegung des SCR-Systems die zentrale Rolle [85].

2.1.3. Entwicklungsmethodik im Bereich SCR

In diesem Abschnitt wird versucht, einen Überblick über den aktuellen Stand der Entwicklungsmethodik im Entwurf von SCR-Systemen für nicht-stationäre Anwendungen zu geben. Dies stellt sich prinzipiell schwierig dar, da die Entwicklung von SCR-Systemen im Wesentli-

chen von den großen OEM-Motorenhersteller bzw. von den Abgassystemherstellern im Auftrag der OEM-Hersteller vorangetrieben wird. In dieser Konstellation werden aus Geheimhaltungsgründen in der Regel keine Berichte über die Entwicklungsmethoden der Unternehmen veröffentlicht. Veröffentlichungen bezüglich der SCR-Systeme beschränken sich im Wesentlichen auf die Vorstellung von fertig entwickelten Lösungen (z.B. [121, 166, 83, 167]).

Die methodischen Entwicklungen im Bereich der SCR-Systementwicklung finden hauptsächlich in der Weiterentwicklung von Strömungssimulationsprogrammen statt, welche eine immer exaktere Berechnung der Vorgänge im SCR-System erlauben. Dies beinhaltet vor allem die integrierte Berechnung von Abgasströmung, HWL-Dosierung und Wandfilmbildung inklusive der chemischen Reaktionen in der Hydrolysestrecke sowie im Katalysator [20, 22, 21, 57, 182].

Im Bereich der Fahrzeugapplikation von SCR-Systemen gibt es Design of Experiments (DoE)-basierte Ansätze, die es erlauben, am Prüffeld automatisiert die optimale Systemkalibration eines SCR-Systems im Zusammenspiel mit der Motorkalibration durchzuführen [88]. Diese Ansätze beziehen sich nicht auf die frühe konzeptionelle Entwicklungsphase von SCR-Systemen, sondern auf die abschließende Phase der Applikation der praktisch fertig entwickelten SCR-Komponenten am Verbrennungsmotor.

Die ICE Strömungsforschung GmbH² bietet eine integrierte Lösung für den Konzeptentwurf von Abgassystemen an. Die Software "DexaSIM" integriert eine speziell auf die Erstellung von Abgassystemen abgestimmte Geometrieerstellung und Strömungsberechnung. Die Geometrie-modellierung geschieht dabei manuell.

Die im Rahmen dieser Arbeit entwickelten Ansätze zur autonomen und virtuellen Entwicklung von SCR-Systemen mit Hilfe von graphenbasierten Entwurfssprachen erlauben eine vollständig automatisierte Auslegung und simulative Validierung von SCR-Systementwürfen [196, 197]. Auf Basis gegebener Anforderungen (Motordaten, 3D-Modell des Bauraumes und der Zielemissionen) findet eine automatisiert ablaufende Geometrieerstellung im gegebenen Bauraum und eine anschließende Strömungssimulation statt. Die Ergebnisse der Strömungssimulation werden hierbei automatisch ausgewertet und können prinzipiell in eine regelbasierte Optimierung des Entwurfes einfließen.

Die in [196, 197] veröffentlichten Inhalte stellen die prinzipielle Funktionsweise der im Rahmen dieser Arbeit erstellten Entwurfssprache zur Entwicklung von SCR-Systemen dar. In den Veröffentlichungen sind die in dieser Arbeit dargestellten Ausführungen zu Ordnungsmechanismen sowie zur Verrohrungssprache allerdings nicht enthalten.

Im Bereich des Entwurfs von Abgasrohrleitungen sind in den meisten CAD-Programmen Assistenzsysteme zu finden. Diese erlauben aber keine Vollautomatisierung. In anderen Anwendungen (vor allem Anlagenbau beziehungsweise in dem dem Anlagenbau ähnlichen Großschiffsbau) kommen Tools zur (teil-)automatisierten Verrohrung zur Anwendung, beziehungsweise befinden sich im Forschungsstadium (siehe auch Kapitel 2.4.2). Diese Anwendungsbereiche weisen aber in der Regel, beispielsweise durch ihre standardisierten Trassenführungen und Verrohrungsregeln [163], andere Anforderungen auf, als der Entwurf eines SCR-Systems in engen und verwinkelten Fahrzeugbauräumen. Die Grenzen zwischen den Anwendungen sind aber fließend.

²www.ice-sf.at

2.2. Ingenieurwissenschaftlicher Entwurf

Die folgenden Abschnitte sollen ausgewählte, für die vorliegende Arbeit relevante, Teile der Entwurfsmethodik darstellen. Dabei kann auf Grund der Diversifikation des Wissensgebietes kein Anspruch auf eine vollständige Abdeckung des gesamten Themenfeldes erhoben werden.

2.2.1. Produktentwurf

Allgemeiner Entwurf

"Design can be viewed as the process of specifying a description of an artifact that satisfies constraints arising from a number of sources by using diverse sources of knowledge."

- aus [117]³.

Das obige Zitat gibt eine allgemeine Definition des Begriffs des Entwerfens wieder, die aber in den Augen des Autors wesentliche Stichworte des wissensbasierten Entwurfs abdeckt. In der Anwendung des computergestützten Entwurfes kann die Aussage: "specifying a description" in doppelter Bedeutung aufgefasst werden: Zum einen als grundsätzliche Notwendigkeit - als die erste zu lösende Aufgabe im computergestützten Entwurf - eine im Computer verarbeitbare Beschreibung eines ingenieurwissenschaftlichen Entwurfes zu definieren. Zum anderen, im weiter gefassten Kontext, kann die Aussage als der Endpunkt des Entwurfsprozesses, nämlich die finale Definition des Produktes, aufgefasst werden.

Der eigentliche Prozess des Produktentwurfes eines konkreten Objektes stellt dann die Synthese eines finalen Entwurfes unter Verwendung der anfangs definierten Beschreibung des zu entwerfenden Objektes dar. Gleichzeitig müssen gegebene Anforderungen (Requirements) und sich im Prozess des Aufbaus des Entwurfsobjektes ergebende Zwangs- und Randbedingungen (Constraints) erfüllt werden.

Über die Strukturierung und Beschreibung des eigentlichen Entwurfsprozesses sind viele verschiedene Herangehensweisen und Vorschläge in der Literatur zu finden. Es kann festgehalten werden, dass es keine generell akzeptierte, vereinheitlichte und in der Praxis anwendbare Entwurfstheorie des ingenieurwissenschaftlichen Entwurfsprozesses gibt [155, 117].

Lawson kommt in [117] zum Schluss, dass eine Vielzahl möglicher Beschreibungen und Wege des Entwerfens existieren: "Hopefully this book has shown that just as there are many ways of designing so there are many ways of describing design." Die Quelle [34] gibt einen weiteren guten Überblick über die Historie und die Inhalte des im Sinne von Lawson weitgefassten Forschungsfeldes der Entwurfsmethodik (Design Research).

Eine oft verwendete und international vielzitierte Quelle der Entwurfsmethodik stellt das Werk [138] von Pahl und Beitz dar. Es zeigt in einer abstrakten Form einen prinzipiell zu verfolgenden Entwurfsprozess, der sehr eng mit einer entsprechenden Richtlinie des Verein Deutscher Ingenieure (VDI) verbunden ist [193]. Der Entwurfsprozess aus [138] umfasst im Wesentlichen die unten dargestellten Schritte, die innerhalb des Entwurfsprozesses in der gegebenen Reihenfolge abgearbeitet werden.

³Das Werk von Lawson weist zwar schwerpunktmäßig Beispiele aus der Architektur aus, in der Einführung stellt Lawson aber den universellen Charakter des Werkes klar: "...this is not a book about architecture, or indeed about any of the products of design. It is a book about design problems...".

Nach Pahl und Beitz kann es in dem Schema auch grundsätzlich zu iterativen Sub-Entwurfsschleifen kommen, die partiell entgegen der nummerierten Ordnung laufen:

1. Festlegen der Anforderungsliste
2. Festlegen der prinzipiellen Lösung
3. Festlegen des vorläufigen Entwurfes
4. Festlegen des endgültigen Entwurfes
5. Festlegen der Produktdokumentation

Der letzte Punkt, die Festlegung der Produktdokumentation, stellt eher eine formale Station des Produktentwurfes dar, da sie nichts mit dem eigentlichen Entwurf des Produktes zu tun hat. Die Quelle [71] verwendet das gezeigte Entwurfsschema nach [138] zur Strukturierung des wissensbasierten Entwurfs eines Kleinsatelliten mit Entwurfssprachen. Über den konkreten Ablauf auf der Ebene einzelner, konkreter Entwurfshandlungen kann die Quelle [138] aber, durch die abstrakte und auf Allgemeingültigkeit abzielende Prozessbeschreibung, keine verbindlichen Antworten geben. Jedoch enthält [138] eine interessante und ausführliche Sammlung für die Konstruktions- und Entwurfsmethodik relevanter Quellen in chronologischer Auflistung.

Folgt man dem oben zitierten Satz aus [117] von Lawson, so kann gemutmaßt werden, dass z.B. auf Grund der Mehrdeutigkeit der geometrischen Beschreibung, wie zusammenfassend von Stiny in [8] dargestellt, ohne Beschränkung der Allgemeingültigkeit keine eindeutige Beschreibung und Abfolge des Entwurfsprozesses, zumindest geometrisch manifestierter Produkte, existiert. In diesem Sinne wird in dieser Arbeit dem Zitat aus [117] gefolgt: "It remains the case that the design process can be learned chiefly through practice and is very difficult to teach well. It is extremely difficult to understand design without actually doing it." Darum soll in den folgenden Kapiteln zunächst der wissensbasierte Entwurf von SCR-Systemen, mit den im weiteren Verlauf dargestellten Entwurfssprachen, praktisch vollzogen und aus der Implementierung Erkenntnisse für den Entwurfsprozess gezogen werden.

Computergestützter Entwurf Mit steigender Komplexität der Entwürfe und dem Streben nach Effizienzsteigerung in der Wirtschaft setzt auch im Produktentwurf der Siegeszug des Computers und entsprechender computergestützter Methoden zur Optimierung und Automatisierung der Arbeitsabläufe ein [138]. Die sich in der Anwendung befindlichen computergestützten Methoden beschränken sich aber im Wesentlichen auf Programme, die zwar Arbeitsabläufe unterstützen, im besten Falle Teilaufgaben automatisieren können, aber keine Vollautomatisierung des Entwurfsprozesses erlauben (siehe Kapitel "Rechnerunterstützung" in [138]).

Ein Haupttrend in der Entwicklung kommerzieller Programme ist das Zusammenwachsen von CAD-Geometriemodellierung (Computer-Aided Design) und Analyseprogrammen (FEM, CFD) sowie der Fertigungsprogrammierung (Computer Aided Manufacturing (CAM)) in ein gemeinsames Computer Aided Engineering (CAE)-Programmpaket mit durchgehendem Datenaustausch. Solche CAE-Systeme bieten im Allgemeinen Programmierschnittstellen, welche die Implementierung einer weiteren maßgeschneiderten Automatisierung von Teilprozessen, wie beispielsweise in [54] dargestellt, erlauben.

Wissensbasierter Entwurf: Formalisierte Entwurfssysteme Unter dem Stichwort des Knowledge Based Engineering (KBE) wird im Allgemeinen das maschinelle Erfassen und Wiederausführbarmachen von Entwurfswissen verstanden [195]. Das KBE stellt ein Spezial-

gebiet der wissensbasierten Systeme (Knowledge-Based Systems) dar, welches sich mit der Verwendung von KI zur allgemeinen Problemlösung befasst [1, 161]. Die Quelle [192] gibt einen weiterführenden und tiefgehenden Einblick über grundlegende Techniken der KI in der ingenieurwissenschaftlichen Anwendung.

Formalisierte Entwurfssysteme, die eine computergestützte, vollautomatisierte Synthese von Produktentwürfen oder zumindest von Teilaspekten des Produktes (beispielsweise die automatisierte Synthese der Geometrie) erlauben, stellen eine Ausprägung des KBE dar. Die Quelle [8] liefert einen Überblick über einige ausgeführte formalisierte Entwurfssysteme, die von der Synthese der Geometrie von Kaffeemaschinen über die Generierung kinematischer Systeme bis hin zur automatisierten Konfiguration mikroelektronischer Schaltkreise reicht.

Die in dieser Arbeit eingesetzten Entwurfssprachen, die eine vollständige maschinelle Automatisierung des Entwurfes erlauben, sind damit in diese Kategorie der formalisierten Entwurfssysteme einzuordnen. In den weiteren Ausführungen zum Stand des Wissens soll daher der Fokus auf formalisierte wissensbasierte Entwurfsprozesse, im Speziellen graphenbasierte Entwurfssprachen, sowie auf die zur Implementierung verwendeten Werkzeuge gelegt werden.

Dem Konzeptentwurf kommt im Entwurfsprozess ein entscheidender Stellenwert zu, da die Entscheidungen in der frühen Phase schon einen großen Teil der später auflaufenden Produkt-(Entwicklungs-)kosten bestimmen [138]. Daher verspricht eine computergestützte Durchführung des Vorentwurfes [155, 180, 171] einen hohen Nutzen für den gesamten Entwurfsprozess. Die Verwendung von Entwurfssprachen erlaubt schon im frühen Vorentwurf Modelle für die (physikalische) Analyse des Entwurfes zu generieren [84] und führt so neben der Beschleunigung durch Automatisierung auch zu einer höherwertigen Validierung des Konzeptentwurfes [160, 157, 26].

Dem eiligen Leser sei empfohlen direkt zu Kapitel 2.3 zu springen. Dort wird das Prinzip des wissensbasierten Entwurfes mit Entwurfssprachen dargestellt. In den direkt folgenden Abschnitten werden ausgewählte, für die Arbeit und für die Implementierung der verwendeten Entwurfssprache relevante Aspekte und Methoden ergänzt.

2.2.2. Modellierung

Modellarchitektur Betrachtet man den industriellen Entwurf von Produkten, so zeigt sich eine große Vielzahl von verschiedenen Softwarelösungen, die ihre jeweiligen Modelle in jeweils eigenen Datenformaten speichern [54, 183]. Diese Datenformate sind darüber hinaus im Allgemeinen proprietärer Natur, was mutmaßlich vor allem auf kommerzielle Interessen der einzelnen Softwarefirmen zurückzuführen ist. Während der in der Entwurfsphase auftretenden Entwurfschleifen [138] müssen diese einzelnen Modelle, auf Grund der mangelnden Interkommunikativität, stets einzeln aktualisiert werden.

Für eine Automatisierung des Entwurfsprozesses müssen aber Änderungen in einem Modell stets auch automatisiert in allen anderen Modellen nachgezogen werden. Dies kann beim Einsatz von N -Modellen im Entwurf entweder über die Verwendung von $\frac{N \cdot (N-1)}{2}$ bidirektionaler Schnittstellen zwischen jedem der N Modelle (siehe Abbildung 2.3 links) oder über den Einsatz eines übergeordneten, vereinheitlichten Datenmodells (Unified Digital Master Model, siehe Abbildung 2.3 rechts) erreicht werden [157, 158]. Die Verwendung eines übergeordneten Datenmodells führt so auf eine Zahl von nur noch N bidirektionaler Schnittstellen [158].

Ein weiteres Argument spricht nach Rudolph [157] für die Verwendung eines einheitlichen Datenmodells: Bei Verwendung von bidirektionalen Schnittstellen zwischen den Einzelmodellen

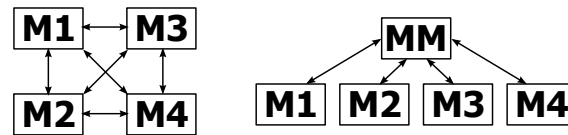


Abbildung 2.3: Gleichberechtigte Modelle $M1$ bis $M4$ mit $\frac{N \cdot (N-1)}{2} = 6$ bidirektionalen Schnittstellen (links). Übergeordnetes Datenmodell MM mit untergeordneten Modellen $M1$ bis $M4$ und $N = 4$ bidirektionalen Schnittstellen (rechts).

müssen Modelle mit unterschiedlichen mathematischen Dimensionen der beschreibenden Parametersätze bidirektional übersetzt werden. Eine umkehrbar eindeutige (bijektive) Propagation von Änderungen zwischen den Modellen existiert nur zwischen Modellen (Räumen) gleicher Dimension [157]. Für den Austausch von Änderungen zwischen unterschiedlich dimensionalen Modellen gibt es nach [157] nur eine konsistente Änderungsrichtung: Eine Projektion von dem höherdimensionalen zum niedrigerdimensionalen Modell. Daraus kann nach Rudolph [157] der Schluss gezogen werden, ein einheitliches übergeordnetes Datenmodell (Unified Digital Master Model) zu verwenden. Aus dem übergeordneten Modell kann dann in Form einer Projektion mit allen Submodellen über eine entsprechende Schnittstelle konsistent kommuniziert werden [157].

Modellierungssprache Zur Modellierung des zu entwerfenden Produktes sowie des Prozesses des Entwerfens bieten sich Modellierungssprachen, wie die Unified Modeling Language (UML) [134], als zentrales Datenmodell an [84]. Die Modellierungssprache UML, ursprünglich in der Entwicklung komplexer Softwaresysteme beheimatet, besitzt Erweiterungsmechanismen, welche eine Anpassung und Anwendung im Bereich des Produktentwurfes unterstützen [146].

Die Arbeit [146] stellt die Eignung des objektorientierten Modellierungsansatzes von ingenieurwissenschaftlichen Objekten unter Verwendung der UML fest. Die objektorientierte UML stellt eine weitere Abstraktionsebene oberhalb der (Objektorientierten-) Programmiersprachen dar und eignet sich damit, abstrakte und komplexe technische Objekte und Entitäten (im UML-Klassendiagramm) sowie zeitliche Abläufe (im UML-Aktivitätsdiagramm) abzubilden und diese Modelle einfach wiederzuverwenden [146].

Die Verwendung von UML zur Modellierung des ingenieurwissenschaftlichen Entwurfes erlaubt daneben auf einer abstrakten Ebene die ebenbürtige Modellierung sowohl von Softwaremodellen als auch von ingenieurwissenschaftlichen Modellen [84]. Die UML verspricht damit eine effektive Unterstützung der Modellierung von formalisierten Entwurfssystemen, welche in sich, mit der Verwendung von KI zur Automatisierung des ingenieurwissenschaftlichen Entwurfes, ebenfalls beide Felder vereinen. Außerdem trägt die Integration von Software- und Produktmodellierung in einer softwaretechnischen Modellierungsebene dem zunehmenden Trend der Darstellung von Produktfunktionen unter Verwendung von Software Rechnung [155].

Eine in [8] von Ward zitierte Faustregel des Konstruierens, "design brackets; buy mechanisms", zeigt beispielhaft die Verwandtschaft dieser in der mechanischen Konstruktion gängigen Herangehensweisen mit einem in [66] postulierten Leitsatz der objektorientierten Programmierung: "Program to an 'interface', not an 'implementation' ". In der (Re-)Kombination einzelner Lösungen beziehungsweise Subsysteme zu einem Produktentwurf sowie in der Integration und Verwendung verschiedener (Analyse-)Modelle spiegelt sich der zentrale Gedanken des System-of-Systems (SoS) [15] direkt in der Struktur und Implementierung von Entwurfssprachen (Kapitel 2.3) wieder [158].

Entwurfsparadigmen: Top-Down und Bottom-Up Die Quellen [4, 140, 110] verwenden eine formale Einteilung des Entwurfsprozesses, unter Verwendung einer SoS-Betrachtungsweise des Entwurfes, in Entwurfsdomänen mit im Voraus bekanntem Verhalten und solchen Domänen, deren Verhalten und Bewertungen a priori unklar sind. Domänen mit im Voraus bekanntem bzw. berechenbarem Verhalten führen auf ein Top-Down-Entwurfsprinzip. Bei der Selektion eines zielführenden Lösungsprinzips beziehungsweise Subsystems kann auf entsprechende analytische Ausdrücke oder auf in einer anderen Weise abgelegten expliziten Vorwissens über den Entwurf und dessen Bewertung zurückgegriffen werden [4, 140]. Ein solcher Top-Down-Entwurfsprozess ist beispielsweise in analytisch gut verstandenen und beschreibbaren (Teil-)Systemen und Teilaufgaben, wie dem Satellitenentwurf [164, 72] oder der Konzeptauslegung von Luftschiffen [159], anzutreffen und deckt sich nach [4] mit der von Pahl und Beitz in [138] dargestellten manuellen Vorgehensweise im Produktentwurf.

Entwürfe, deren Verhalten nicht im Voraus bestimmt werden kann, führen auf einen Bottom-Up-Entwurfsprozess. Dabei ist im Prozess des Entwerfens beispielsweise die Lösung eines Feldproblems notwendig, welches auf Grund der hohen Komplexität oder Nichtlinearität der beteiligten Physikdomänen (Strömungsturbulenz, große Verformungen, Materialversagen,...) nicht analytisch vorhergesagt werden kann. Damit muss der Entwurf jeweils explizit generiert und analysiert werden [4, 140]. Diese Kategorie entspricht nach Quelle [110] formal einem Bottom-Up-Entwurf, in dem verschiedene Entwürfe ohne explizites Wissen über die finale Bewertung synthetisiert und dann in den Prozessketten bewertet werden. Solche Bottom-Up-Prinzipien ergeben sich zum Beispiel in der Synthese von Strommasten [3] und im automatisierten Entwurf von Bauteilen aus Faserverbundwerkstoffen [111], da dort die expliziten Berechnungsvorschriften zur a priori Bewertung des Gesamtsystems unbekannt sind.

2.2.3. Standardisierung

Die Standardisierung und Normierung, als Vereinheitlichung von Lösungen und Lösungsprinzipien [138], spielt eine historisch etablierte, wichtige Rolle im Ingenieurentwurf [138, 189]. Prozesse, Bauteile und Baugruppen werden aus unterschiedlichen Gründen standardisiert. Die folgende schlaglichtartige Auflistung folgt einer von der International Organization for Standardization (ISO) [128] veröffentlichten Broschüre⁴ [93]:

- **Innovation:** Die Standardisierung stellt eine gemeinsame Sprache und Metrik zur Verfügung, um Produkte schneller auf den Markt zu bringen und deren Kompatibilität sicher zu stellen.
- **Kostenreduktion:** Die Standardisierung führt zu einer Optimierung der Entwurfszyklen und damit zu einer Reduktion der Produktentwicklungs- und Fertigungskosten.
- **Produktsicherheit:** Normen stellen die technische Basis zur Einhaltung gesetzlicher Vorschriften und Regelungen dar und vermindern dadurch das Risiko von Unfällen und Unglücken.
- **Globaler Marktzugang:** Standardisierung verhindert Handelsbarrieren, eröffnet neue Märkte, erweitert bestehende Märkte und vereinfacht den Handel.

⁴Die zitierte Quelle stützt sich auf diverse Untersuchungen, durchgeführt sowohl von der ISO als auch von Mitgliedsorganisationen. Für den Zweck der vorliegenden Arbeit, der keine detaillierte Untersuchung der Produktstandardisierung umfasst, soll die bewusste schlaglichtartige Wiedergabe der wichtigsten Argumente, auch ohne jeweilige explizite Begründungen durch wissenschaftliche Einzelquellen, genügen.

- Risikomanagement: Normen helfen Risiken zu bewerten und zu reduzieren sowie Unsicherheiten zu bewältigen.
- Umwelteinfluss: Standardisierung vermindert negative Umwelteinflüsse und verbessert die Umweltverträglichkeit von Produkten.
- Qualitätssicherung: Standardisierung führt zu erhöhter Kundenzufriedenheit und unterstützt eine kontinuierliche (Produkt-)Verbesserung.
- Kunden: Die Standardisierung stellt ein Rahmenwerk für kommerzielle Verträge und zur Konfliktlösung zwischen Unternehmen zur Verfügung.
- Energieeffizienz: Standardisierung erleichtert einen rationellen Umgang mit Energie und eröffnet Märkte für energieeffiziente Technologien.
- Soziale Verantwortung: Überführt ethische Regeln und Handlungsweisen in konkrete Handlungsanweisungen.

Darüber hinaus führt eine Standardisierung integrierter, komplexer Subsysteme und Komponenten auf die im Automobilbau immer populärer werdenden Baukastensysteme [169]. Die Standardisierung und Normierung innerhalb des globalen Marktgeschehens wird im Wesentlichen von der ISO und ihrer nationalen Mitgliedsverbände, für Deutschland das Deutsches Institut für Normung (DIN) [104], organisiert und ausgeführt. Die Standardisierung und Normung stellt damit ein globales Rahmenwerk und eine gemeinsame Sprache für Ingenieure dar um technische Informationen auszutauschen und vereinheitlichen zu können [93].

Standardisierungsregeln und Normen führen implizit zu einer Reduktion der möglichen Entwurfsvarianten und Vielfalt. Nach [138] ist diese Reduktion der Entwurfsvarianten durch Normierung aber nicht gleichzusetzen mit einer Einschränkung der Lösungsvielfalt⁵, so dass trotz Standardisierung einzelner Teile und Größen im Entwurf eine optimale Lösung sollte gefunden werden können.

2.2.4. Entwurfsbewertung

Nach [138] kommt der Bewertung eines Entwurfes eine zentrale Rolle zu. Die in einem systematischen Produktentwicklungsprozess gewonnenen möglichen Lösungen müssen nach [138] bewertet und auf Basis dieser Bewertung die zielführenden Lösungsvarianten identifiziert werden. In [138] sind einige der in der Praxis gebräuchlichen Methoden zur Entwurfsbewertung dargestellt. Die dargestellten Methoden erlauben zwar die Aggregation von Sub-Bewertungen zu einer Gesamtbewertung des Entwurfes, die verwendeten Bewertungsskalen und Gewichte der einzelnen Subbewertungen werden aber willkürlich vom Entwerfer gewählt.

Dieser Mißstand der willkürlichen und damit inkonsistenten Bewertung wird von Rudolph in [154] aufgegriffen und zur Behebung ein generisches Entwurfsbewertungsverfahren vorgeschlagen. In [154] wird zu Beginn eine eindeutige und reproduzierbare Bewertung, die gerade nicht auf willkürlichen Bewertungsmästben und Bewertungsgewichten basieren darf, welche potentiell auf redundante Bewertungsaggregationen fhren wrde, gefordert. Ausgehend von

⁵Zitat aus [138] Kapitel "Normengerecht": "Dabei steht die lsungsbeschrnkende Zielsetzung der Normung nicht im Gegensatz zu einer Varianten anstrebenden methodischen Lsungssuche, da die Normung sich im wesentlichen auf die Festlegung einzelner Elemente, Teillsungen, Werkstoffe, Berechnungsverfahren, Prfvorschriften und dergleichen konzentriert, die Lsungsvielfalt und Lsungsoptimierung hingegen durch Kombination beziehungsweise Synthese bekannter Elemente und Gegebenheiten erreicht werden."

diesen Forderungen wird die Äquivalenz von Entwurfsbeschreibung und Entwurfsbewertung unter Verwendung des Pi-Theorems [33] postuliert [154]:

”Jede im Sinne des Pi-Theorems minimale Beschreibung ist eine Bewertung.”

Das Pi-Theorem ordnet einer dimensionshomogenen Beschreibungsfunktion $f(x_1, \dots, x_i, \dots, x_N)$ eines Systems eine Bewertungsfunktion $F(\Pi_1, \dots, \Pi_j, \dots, \Pi_M)$ mit $M = N - R$ zu, die aus den dimensionslosen Bewertungskriterien Π_j gebildet werden [155]. Der Parameter R gibt dabei den Rang der sogenannten Dimensionsmatrix an, welche den Variablen x_i (Zeilen) ihre entsprechenden physikalischen Basiseinheiten wie Länge, Zeit, usw. zuordnet (Spalten).

Eine konsistente, physikalische Entwurfsbeschreibung, die implizit dimensionshomogen ist, stellt damit eine Beschreibung im Sinne von Rudolph dar und kann direkt zur Ableitung entsprechender Bewertungsgrößen des jeweiligen Entwurfes unter Verwendung des Pi-Theorems verwendet werden. Dies macht ein im Entwurfsprozess (unter Verwendung formalisierter Entwurfssysteme) automatisiert synthetisiertes Gleichungssystem besonders nützlich. Dieses Gleichungssystem stellt nach [155] den Ausgangspunkt der Anwendung des Pi-Theorems als generisches Entwurfsbewertungsverfahren dar. Die Anwendung des Pi-Theorems lässt sich prinzipiell ebenfalls automatisieren. Dieser Gedanke führt im Weiteren auf die Verwendung eines Lösungspfadgenerators im Entwurf mit Entwurfssprachen (siehe Kapitel 2.3.6), welcher eine automatisierte Behandlung von (analytischen) Gleichungs- und Zwangsbedingungssystemen erlaubt [155, 159].

2.2.5. Entwurfsraum

Als Entwurfsraum wird im Rahmen formalisierter Entwurfssysteme der Raum der mit dem Entwurfssystem erzeugbarer formal gültiger⁶ Entwürfe bezeichnet [8, 3, 110]. Die so erzeugbaren Entwürfe sind aber nicht zwingend mit den gegebenen Anforderungen verträglich [8]. Mit Hilfe dieser Entwurfssysteme kann dann im Entwurfsraum systematisch und computergestützt nach zielführenden Entwürfen, welche gegebene Anforderungen und Zielkriterien erfüllen, gesucht werden. Dieser Prozess wird als *Exploration des Entwurfsraumes* bezeichnet [67, 155].

Diese Definition des Entwurfsraumes, als Menge der mit einem System erzeugbaren Entwürfe, deckt sich mit der Definition des impliziten Entwurfsraumes von Woodcraft in [207]:

”The former (implicit design space) is the graph of all possibilities engendered by the codification system used.”

Woodcraft nimmt dabei eine Unterscheidung zwischen dem impliziten Entwurfsraum, welcher implizit in dem die Entwürfe erzeugenden System steckt und dem expliziten Entwurfsraum, welcher nach Woodcraft die Menge der bereits bekannten beziehungsweise ”besuchten”, sinnvollen Entwürfe beschreibt. Im weiteren Verlauf soll bei alleiniger Verwendung des Begriffes ”Entwurfsraum” der implizite Entwurfsraum gemeint sein.

2.2.6. Constraint-Satisfaction-Problem

Ein Constraint (Zwangsbedingung) kann abstrakt als eine Einschränkung eines Raumes von Möglichkeiten dargestellt werden [49]. Das in der Mathematik bekannte Constraint Satisfaction Problem bezeichnet das Auffinden derjenigen Konfigurationen im Raum aller kombinatorisch möglichen Konfigurationen, welche mit den gegebenen Constraints verträglich sind [49]. Das

⁶Bei Verwendung von Entwurfsgammatiken beziehungsweise formaler Sprachen entspricht dies ”syntaktisch” korrekten Entwürfe.

Constraint-Satisfaction-Problem ist formal als $\langle X, D, C \rangle$ definiert. $X = x_1, \dots, x_i, \dots, x_N$ sei dabei eine Menge diskreter Variablen mit Wertemengen $D = D_1, \dots, D_i, \dots, D_N$. X und D bilden dabei den genannten Raum aller kombinatorischen Möglichkeiten [174]. Die Menge C enthält alle auf X definierten Constraints [174].

Die Verwendung von Constraints [117], beziehungsweise die Modellierung [171, 192, 159] des Entwurfsproblem als Constraint-Netzwerk, ist eine vor allem im Konzeptentwurf übliche Vorgehensweise. Bei der Verwendung von Constraints im Entwurf werden aber im Allgemeinen auch arithmetische Zwangsbedingungen in Form von (physikalischen) (Un-)Gleichungen verwendet und damit die in der ursprünglichen Definition eingeführte Beschränkung auf diskrete Variablen aufgegeben (siehe Beispiele in [171, 192, 159]).

Die sukzessive Eingrenzung des Entwurfsraumes über Constraints kann auf eine nicht-leere Untermenge des Entwurfsraumes führen, welche dann als Endpunkt des Entwurfsprozesses die zielführenden Konfigurationen des Produktes repräsentiert. Dieses Entwurfsparadigma stellt damit eine deklarative Herangehensweise an das Entwurfsproblem dar.

Wie in Kapitel 2.3.6 dargestellt enthält der Entwurfscompiler, der die in dieser Arbeit entwickelten Entwurfssprachen verarbeitet, einen Lösungspfadgenerator zur automatisierten Auflösung solcher Zwangsbedingungssysteme im Entwurfsprozess. Dieser Lösungspfadgenerator wird darüber hinaus im Rahmen dieser Arbeit weiterentwickelt (Kapitel 5.1).

2.3. Entwurfssprachen

In den folgenden Abschnitten soll die Entwurfsmethode der Entwurfssprachen dargestellt werden. Zunächst wird eine theoretische Motivation der Methode gegeben. In den weiteren Abschnitten werden die Eigenschaften der Entwurfssprachen sowie die zur Erstellung und Verarbeitung notwendigen Komponenten erläutert.

2.3.1. Motivation

Die Motivation der Entwurfssprachen im folgenden Abschnitt lehnt sich in verkürzter Form der Darstellung von Rudolph in [155] an. Rudolph setzt sich in [155] zunächst mit der Beschreibung des Entwurfes auseinander und nimmt folgende Einteilung der Begriffsgebiete des Entwurfsprozesses vor [155]:

- *Wissen*: Konsistent, beweisbar oder durch Experimente messbar und reproduzierbar
- *Können*: Nicht beweisbare, einfach handhabbare, vernünftige Handlungsweisen ohne explizite Ableitung aus einer Theorie
- *Glauben*: Ideale zur Orientierung, Werterhaltung und Wertbestimmung

Es sei angemerkt, dass beispielsweise die Physik, welche auf mathematisch beweisbaren Theorien und Modellen beruht, in das Gebiet des *Wissens* einzuordnen ist, da die Modellbildung in der Physik durch Experimente und die Logik der Mathematik abgesichert ist [155]. Im *Können* finden sich nach Rudolph [155] praktische und "vernünftige" aber nicht explizit beweisbare Handlungsanweisungen des Entwurfes, beispielsweise fertigungsgerecht, wartungsgerecht und recyclinggerecht wieder. Die Begrifflichkeiten im Bereich *Glauben* stützen sich nach [155] auf freiwillig akzeptierte philosophische Ideen und Ideale.

Der in Abbildung 2.4 dargestellte iterative Entwurfsprozess überstreicht nach Rudolph, ausgehend von einer (Produkt-)Idee, die im Begriffsgebiet des *Glaubens* angesiedelt ist, die oben genannten, ineinander eingebetteten Begriffsgebiete. Die Synthese und Analyse eines konkreten Entwurfes stößt dann bis in das Gebiet des *Wissens* vor, was beispielsweise die Dimensionierung und Bewertung eines Entwurfes mit Hilfe physikalischer Gleichungen umfasst.

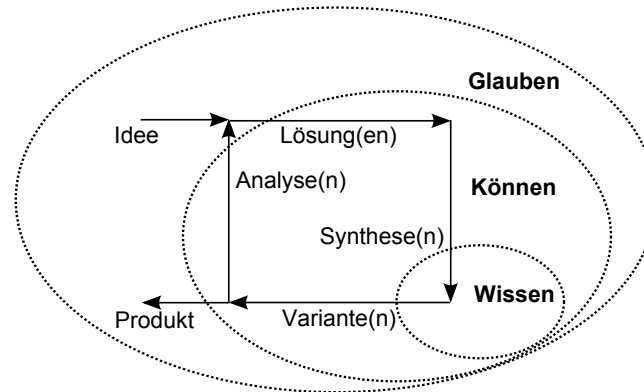


Abbildung 2.4: Entwurfsprozess eingebettet in die Begriffsgebiete *Glauben*, *Können* und *Wissen* nach Rudolph aus [155].

Für eine computergestützte Automatisierung des Entwurfsprozesses, mit dem Ziel eines Entwurfssystems das KI aufweist, leitet Rudolph in [155] die Forderung nach einem Weltmodell beziehungsweise einer universellen Beschreibung des Entwurfes ab, welche die oben genannten Begriffsgebiete vollständig erfassen muss. Im Ingenieurentwurf liegen im Allgemeinen verschiedene, miteinander unverträgliche Modellbeschreibungen (siehe Kapitel 2.2.2) sowie miteinander inkompatible Modelle bezüglich parametrischer und topologischer Lösungsvarianten [155] vor. Die Abwesenheit einer universellen Beschreibung stellt nach [155] auch der Hauptgrund für das bisherige Fehlen einer integrierten und durchgängigen Softwareunterstützung im Konzeptentwurf eines Produktes dar.

Sogenannte graphenbasierte Entwurfssprachen sind nach Rudolph eine mögliche Lösung des Beschreibungsproblems [155]: Eine graphenbasierte Entwurfssprache stellt eine künstliche Sprache dar, welche die zu beschreibenden Entitäten als so genannte Vokabeln sowie ein Regelwerk, die Grammatik, zur Kombination dieser Vokabeln zu Sätzen, welche syntaktisch gültige Entwürfe repräsentieren, enthält. Diese formalen Sprachen imitieren damit das Vorbild natürlicher Sprachen zur Beschreibung von Entitäten und logischen Zusammenhängen [114]. Die logische Abfolge der Regeln zum inkrementellen Aufbau des Gesamtentwurfs, repräsentiert in Form des Entwurfsgraphen, wird als Produktionssystem bezeichnet (beispielsweise als Aktivitätsdiagramm der Modellierungssprache UML).

Eine graphische Repräsentation der Vokabeln (beispielsweise als Klassen der objektorientierten Modellierungssprache UML) als Knoten eines Graphen erlaubt die Modellierung topologischer Beziehungen zwischen den Vokabeln [3]. Eine Entwurfssprache vereint damit sowohl die Beschreibung (Vokabeln) als auch die Regeln, sowie die Regelabfolge (Produktionssystem) zum sinnvollen Zusammensetzen der beschreibenden Vokabeln zum Entwurfsgraphen. Dieser stellt eine abstrakte Repräsentation eines Produktes dar. Eine Entwurfssprache kann so formal den Entwurfsprozess beziehungsweise das Entwurfswissen vollständig abbilden (siehe Abbildung 2.5 links). Die Repräsentation des Produktentwurfes als Vokabeln im Computer, als auch die logische Abbildung der Prozesse des Produktentwurfes in den Regeln der Grammatik und des

Produktionssystem, bilden zusammen die Grundlage für eine wissensbasierte, automatisierte und computergestützte Erstellung von Produktentwürfen mit Entwurfssprachen [155].

Die Entwurfssprachen stellen damit eine verallgemeinernde Erweiterung von Entwurfsgrammatiken dar [4]. Auch in anderen Anwendungsbereichen kommen Entwurfsgrammatiken beziehungsweise formale Sprachen zum Einsatz: So werden sogenannten L-Systeme zur Synthese von Modellen biologischer Objekte, zum Beispiel Pflanzen, eingesetzt [144]. Dabei wird der Aufbau von Pflanzen (Stamm, Äste, Astverzweigungen, Blätter) in Form einer Zeichenkette gespeichert, welche regelbasiert auf- und umgebaut werden kann. Diese Zeichenkette muss zur visuellen Darstellung der Pflanze in die entsprechende graphische Repräsentation übersetzt werden. Im Falle der L-Systeme wird diese Übersetzung *Turtle Interpretation* genannt [144]. Solche L-Systeme sind besonders geeignet zur Erzeugung und Darstellung selbstähnlicher Strukturen, wie zum Beispiel der fraktalen Kochkurve [144]. Die Regeln zum Aufbau der Systeme können stochastischer Natur sein oder zusätzliche parametrische Variablen, zum Beispiel zur Variation des Winkels von Verzweigungen, enthalten, deren Werte wiederum zufällig sein können [144].

Die zeichenkettenbasierten Entwurfsgrammatiken sind aber schwierig auf den Entwurf von Produkten zu übertragen, da Produktentwürfe im Allgemeinen topologische Ringstrukturen, beispielsweise in der Modellierung der Geometrie eines Gittergerüsts, aufweisen [2]. Solcherlei topologische Ringstrukturen können generisch in einem Graphen abgebildet werden [2, 155].

In [8] stellt Cagan einige Anwendungen von *Shape Grammars* vor. Diese sind Formgrammatiken zur gezielten Synthese und Manipulation der Geometrie von ingenieurwissenschaftlichen Produkten. Die Regeln sowohl zur topologischen als auch zur parametrischen Manipulation sind dabei direkt auf der zu manipulierenden Form definiert. Im Artikel von Cagan in [8] wird die Verwendung von graphenbasierten Grammatiken zur Erweiterung der Formengrammatiken um funktionelle Aspekte des jeweiligen Entwurfes diskutiert und in der Implementierung der topologischen Repräsentation der geometrischen Objekte verwendet.

Für die Beschreibung der Regeln wird in den in [8] dargestellten Beispielen ein graphisches "if \rightarrow then"-Schema verwendet, welches auch in den Entwurfssprachen eingesetzt wird (siehe Abschnitt 2.3.4). Bei der Ausführung wird ein auf der linken Seite der Regel definiertes Muster im Entwurfsgraph gesucht. Wird dieses Muster im Entwurfsgraph gefunden, so wird es durch das auf der rechten Seite der Regel definierte Muster ersetzt. Dabei können einzelne Knoten des Musters der linken Seite auch unverändert in der rechten Seite übernommen werden.

Die dargestellten Formgrammatiken sind, entsprechend ihrem Namen, stark auf die Geometriegenerierung in einer spezifischen Anwendung ausgerichtet [4]. Sie vollziehen damit nicht den oben dargestellten verallgemeinernden Abstraktionsschritt der Entwurfssprachen, in denen beliebige Entitäten durch die Vokabeln repräsentiert werden können. Dies impliziert die Repräsentation der geometrischen Elemente auf gleicher Ebene mit nicht geometrischen Elementen anderer Domänen (zum Beispiel physikalische oder funktionale Eigenschaften). Eine Entwurfssprache kann auch gänzlich ohne Geometriemodellierung auskommen und beispielsweise rein abstrakte Informations- oder Regelnetzwerke erzeugen und repräsentieren.

2.3.2. Entwurfscompiler und Prozessketten

Die in den vorherigen Abschnitten entwickelten Gedanken zur Formalisierung des Produktentwurfes unter Verwendung von Entwurfssprachen, vor allem mit der Integration von Beschreibung (Vokabeln), Regeln und Regelabfolge (Produktionssystem) in einem universellen Datenmodell, erlauben eine integrierte und durchgängige Nachbildung des Entwurfsprozesses

auf abstrakter Ebene. Die Verwendung der UML als quelloffene, erweiterbare und standardisierte Modellierungssprache verspricht eine langfristig (wieder)verwendbare Speicherung des Entwurfswissens. In Analogie zu den im vorherigen Abschnitt dargestellten L-Systemen sollte die abstrakte Beschreibung des Produktes (Entwurfsgraph) in ingenieurwissenschaftliche Modelle übersetzt werden. Die Erstellung dieser Modelle kann verschiedenen Zwecken dienen:

- CAD-Modelle: Einbauuntersuchung, Herstellung, Visualisierung,...
- CFD-Modelle: Strömungsanalyse, Wirkungsgrad, Druckverlust,...
- FEM-Modelle: Bauteilbelastungen, Haltbarkeit, Schwingungsverhalten, Lebensdauer,...
- Regelungsmodelle: Auslegung und Simulation elektrischer Komponenten des Produktes...

Obige Auflistung beispielhafter *Prozessketten* stellt nur eine kleine Auswahl möglicher Zielmodelle dar. Auf Basis der Ergebnisse der Analysen und Simulationen, unter Verwendung der abgeleiteten Modelle, können wiederum Änderungen des Entwurfes getriggert werden, was geschlossene Optimierungsschleifen zulässt. Bei Verwendung der UML zur Repräsentation des universellen Datenmodells drücken sich die Ableitungen der analyseprogrammspezifischen Modelle als sogenannte Model-to-Model-Transformationen aus. Diese können als Analogon zur automatisierten Codegenerierung in der Model-Driven-Architecture im Bereich des Softwareengineering [84] verstanden werden.

Analog zur Computer-Programmierung wird bei der Verwendung der Entwurfssprachen ein sogenannter Entwurfscompiler eingesetzt [4]. Dieser Entwurfscompiler umfasst im wesentlichen zwei logische Schritte [4]:

1. Die Ausführung der Regeln zum automatisierten Aufbau des Entwurfsgraphen als abstrakte Beschreibung des Produktentwurfes.
2. Die Ausführung von Plug-Ins zur Durchführung der Model-to-Model-Transformationen für die Bedienung der, in den sogenannten Prozessketten angeschlossenen, (Analyse-) Programme inklusive der Rückführung der (Analyse-)Ergebnisse in den Entwurfsgraphen.

Die Abbildung 2.5 zeigt eine solche Informationsarchitektur mit der Abbildung des Entwurfswissens in einer Entwurfssprache (Vokabeln, Regeln, Produktionssystem) und der Verarbeitung dieser Entwurfssprache im Entwurfscompiler zum Aufbau des Entwurfsgraphen und zur Bedienung der Prozessketten (von links nach rechts).

Für die vorliegende Arbeit wurde der Entwurfscompiler 43V2 [90] eingesetzt. Dieser verwendet die UML als Modellierungssprache und bietet durch die Implementierung in der Integrated Development Environment (IDE) Eclipse die Möglichkeit auch Java-Code, der auf den Entwurfsgraphen wirkt, zur Laufzeit der Entwurfssprache auszuführen.

Der Entwurfscompiler 43V2 stellt einen Lösungspfadgenerator (siehe Kapitel 2.3.6) zur automatisierten Verarbeitung von Zwangsbedingungssystemen und symbolischen Gleichungssystemen zur Verfügung. Die Gleichungen werden in einem Computeralgebrasystem, entlang der im Lösungspfadgenerator bestimmten Sequenz, direkt gelöst [209].

Auch im Bereich der Entwurfskompilation finden sich in der Literatur alternative Implementierungen [201, 8]. Diese sind aber, analog zur Darstellung von Entwurfssprachen in Abschnitt 2.3.1, auf spezifische Anwendungen zentriert und weisen nicht den Abstraktionsgrad und die generische Einsetzbarkeit des Entwurfcompiler 43V2 auf [4].

Der Compiler 43V2 erlaubt es, anwendungsunabhängige graphenbasierte UML-Modelle regelbasiert zu manipulieren und in andere Modelle oder in Programmcode zu übersetzen [90].

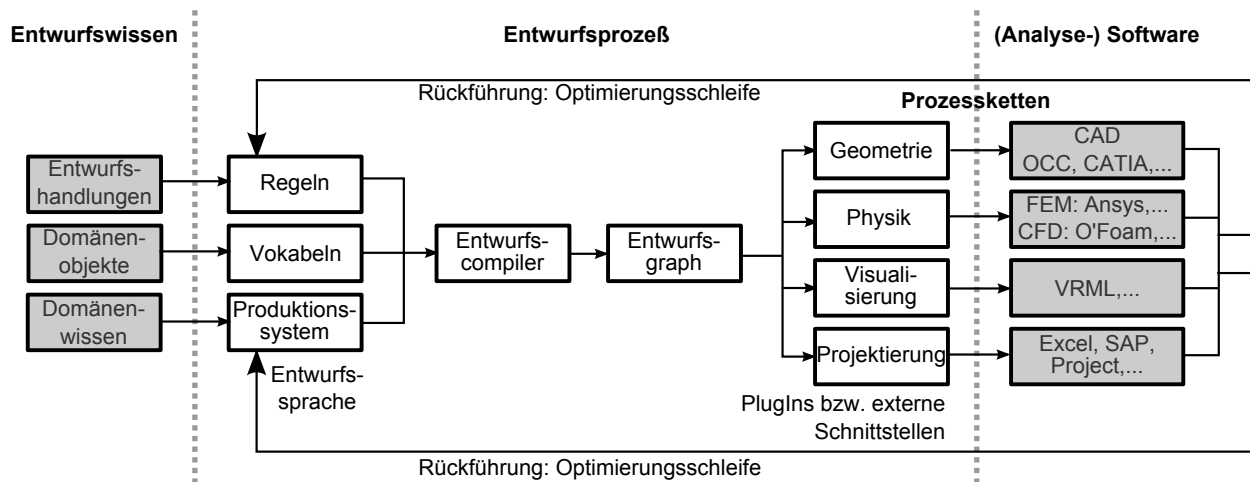


Abbildung 2.5: Informationsflüsse im Entwurfsprozess mit Entwurfssprachen [155, 4, 71].

Externe Programme der Prozessketten sind über leicht zu erweiternde Plug-In-Mechanismen angeschlossen. Die Ergebnisse der externen Programme werden wieder in das zentrale Datenmodell zurückgeführt und erlauben so die Implementierung geschlossener Optimierungsschleifen.

2.3.3. Vokabeln (UML-Klassendiagramm)

Das zu modellierende Produkt wird zunächst topologisch in elementare Entitäten (Bauteile, beschreibende Begrifflichkeiten,...) zerlegt. Diese Entitäten bilden das Vokabular der Entwurfssprache, im Falle der Modellierung mit UML werden diese Entitäten als Klassen abgebildet [84].

Die Klassen (Vokabeln) und die Assoziationen (Beziehungen und Abhängigkeiten) zwischen den Vokabeln bilden das UML-Klassendiagramm. Dieses stellt ein aus Klassen⁷ gebildetes Abbild der abstrahierten Struktur des zu entwerfenden Produktes dar [84]. Topologische Strukturen werden dann isomorph im Klassendiagramm, das ein Graph darstellt, abgebildet [3]. Parametrische Beschreibungen der Klassen können in sogenannten Attributen modelliert werden. Die Attribute einer Klasse stellen die Definition einer (Objekt-)Variablen und deren Datentypen dar [133]. Die Abbildung 2.6 zeigt das Beispiel des Klassendiagrammes eines Motors mit nachgeschaltetem Katalysator.

Die Klassen können in Vererbungshierarchien stehen und so von Elternklassen Attribute und Assoziationen erben, was eine redundanzfreie Weitergabe von Eigenschaften zwischen Klassen erlaubt und eine Modellbildung auf verschiedenen Detaillierungsstufen generisch unterstützt. Abstrakte Klassen sind Klassen die nicht instanziiert werden können und lediglich als Elternklassen innerhalb einer Vererbungshierarchie Eigenschaften an nicht-abstrakte, instanziiierbare Klasse weitergeben können [133].

Modellierung von Constraints Wie in den Abschnitten 2.2.6 und 2.3.6 dargestellt, unterstützt der verwendete Entwurfscompiler 43V2 die Verarbeitung von Constraints. Algebraische Constraints werden als symbolische Gleichungen jeweils in einer Klasse hinterlegt. Gleichungen

⁷Klassenbegriff aus der objektorientierten Programmierung: Klassen werden als Objekttypen aufgefasst und bilden so eine Schablone für einander (strukturell) ähnliche Objekte.

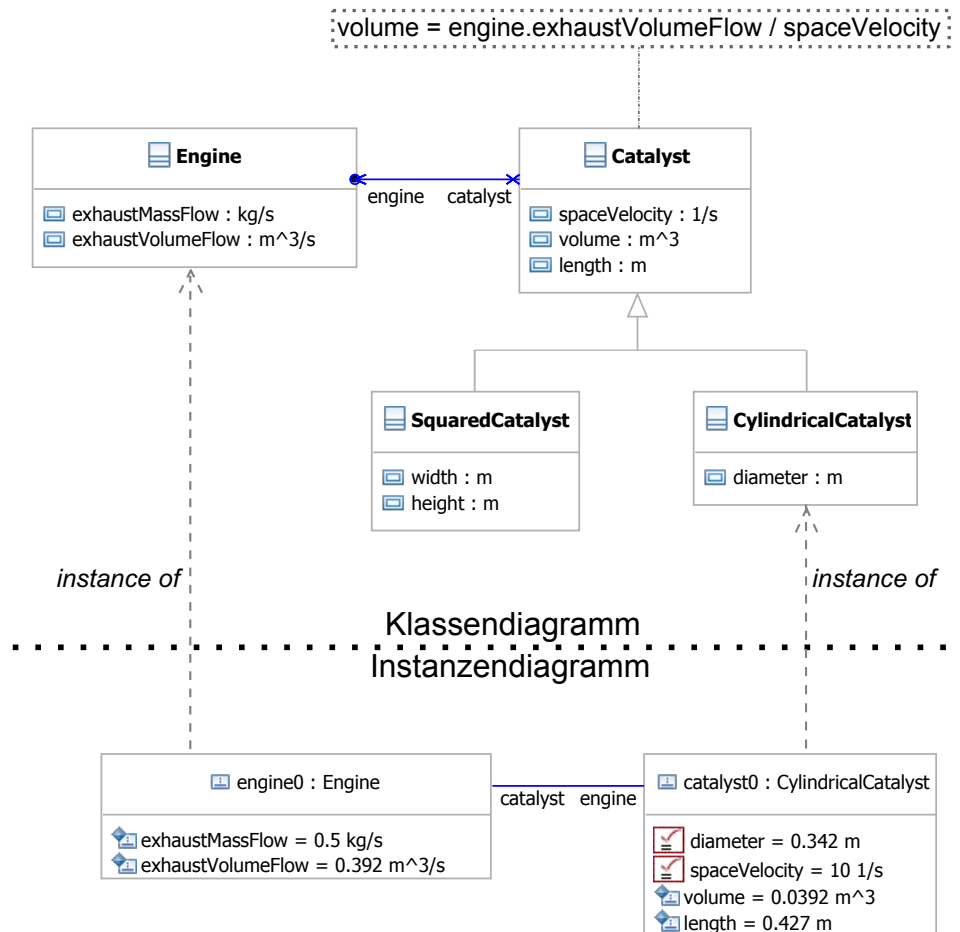


Abbildung 2.6: UML-Klassendiagramm und UML-Instanzendiagramm mit angedeuteter analytischer Gleichungsdefinition zur Berechnung des Katalysatorvolumens aus der Raumeschwindigkeit und dem Abgasvolumenstromes des Motors.

können Variablen der Klasse, in der sie hinterlegt werden, sowie Variablen der über Assoziationen bekannt gemachten Klassen des Klassendiagramms umfassen [17].

Die Abbildung 2.6 zeigt eine solche Gleichungsdefinition sowie die Assoziationen schematisch. Entsprechend Kapitel 2.3.6 müssen für die automatisierte Verarbeitung des Gleichungssystems im Allgemeinen die Variablen mit fest vorgegebenen Werten markiert werden, dies kann in der UML mit sogenannten Stereotypen umgesetzt werden [17, 133].

Entwurfsgraph (UML-Instanzendiagramm) Die Klassen können zu konkreten Objekten instanziiert werden. Bei der Instanziierung werden Objekte einer Klasse als sogenannte Instanzen erzeugt [133]. Dabei wird das Objekt mit einem eindeutigen Instanzennamen und die als Attribute der Klasse definierten Variablen mit konkreten Werten versehen [133]. Der Schritt der Instanziierung ist in Abbildung 2.6 von oben nach unten angedeutet.

Die Instanzen im Klassendiagramm assoziierter Klassen können über sogenannte Links auf Instanzebene entsprechend verbunden werden. Der so entstehende Objektgraph bildet das sogenannte Instanzendiagramm. Im Anwendungsgebiet der Entwurfssprachen wird das in der Ausführung der Entwurfssprache entstehende Instanzendiagramm als *Entwurfsgraph* bezeichnet. Aus diesem digitalen Datenmodell können dann domänenspezifische Modelle (CFD, FEM, CAD,...) automatisiert abgeleitet werden.

2.3.4. Regeln

Die Regeln zur Manipulation des Entwurfsgraphen bestehen entweder aus graphisch formulierten Regeln [4], wie in Abschnitt 2.3.1 angedeutet und in Abbildung 2.7 dargestellt, oder aber aus Java-Regeln, die beim Aufruf Java-Code zur Manipulation des Entwurfsgraphen ausführen [146]. Die Java-Regeln können darüber hinaus den Aufruf anderer Programme oder den Datenaustausch mit dem Dateisystem beinhalten.

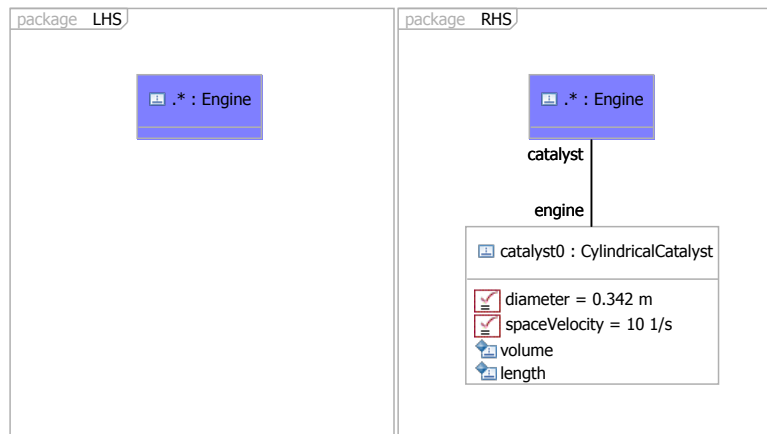


Abbildung 2.7: Graphische Regel mit Konditional-Teil (links) in dem ein Engine-Objekt im Entwurfsgraph gesucht wird. Im Generations-Teil (rechts) wird dem Engine-Objekt ein Katalysator-Objekt hinzugefügt.

Die graphisch formulierten Regeln bestehen zum einen aus einem Konditional-Teil (*if*, linke Seite, LHS), der ein im Entwurfsgraph zu suchendes Muster enthält, zum anderen aus einem Generations-Teil (*then*, rechte Seite, RHS) der den Zustand des auf der linken Seite gesuchten Musters des Entwurfsgraphen nach Ausführung der Regel definiert [4].

In Abbildung 2.7 ist eine graphische Regel dargestellt, welche zu einem *Engine*-Objekt im Entwurfsgraphen einen zylindrischen Katalysatorträger hinzufügt. Die Deklaration der Variablen *spaceVelocity* und *diameter* mit einem Stereotyp (rotes Kästchen mit Gleichheitszeichen) ist zu erkennen und bedeutet, dass diese beiden Variablen in der Lösungspfadsuche als eine der vorgegebenen Größen des Gleichungssystems behandelt werden sollen (siehe auch Kapitel 2.3.6).

2.3.5. Produktionssystem (UML-Aktivitätsdiagramm)

Das Produktionssystem dient der Modellierung der logischen Abfolge der Regeln zum Auf- und Umbau des den Entwurf repräsentierenden Entwurfsgraphen. Das Aktivitätsdiagramm beginnt und endet jeweils an einem *Start*- und *End*-Knoten. Dazwischen kann in Form eines Kontrollflusses die Abfolge von Entscheidungsknoten, (Java-)Regeln, Schnittstellenaufrufen (Plug-In Aufruf des Entwurfscompiler) und Sub-Aktivitäten, hinter denen wiederum hierarchisch verdeckte, eigenständige Aktivitätsdiagrammen stehen, definiert werden [146]. Das Produktionssystem dient damit der Modellierung von Entwurfsprozessen als sequenzielle Abfolge von Teilprozessen und Einzelregeln.

Abbildung 2.8 zeigt ein Aktivitätsdiagramm mit Entscheidungsknoten (Raute). Dabei wird in der dem Entscheidungsknoten zugeordneten Regel *SearchCatalystShape* im Entwurfsgraph nach einer Variablen, im Beispiel ein Katalysatorformparameter, gesucht. Der Wert dieser Variablen geht in einen logischen Ausdruck des Entscheidungsknotens ein. In Abhängigkeit des Ergebnis

der Auswertung des logischen Ausdrucks im Entscheidungsknoten wird eine der beiden dem Knoten nachfolgende Regel aufgerufen. Dabei wird im Beispiel eine der möglichen Katalysatorformen (zylindrisch oder rechteckig), entsprechend dem Wert des Katalysatorformparameters, im Entwurfsgraph eingebaut. Die Regel für den Einbau des zylindrischen Katalysators ist in Abbildung 2.7 dargestellt.

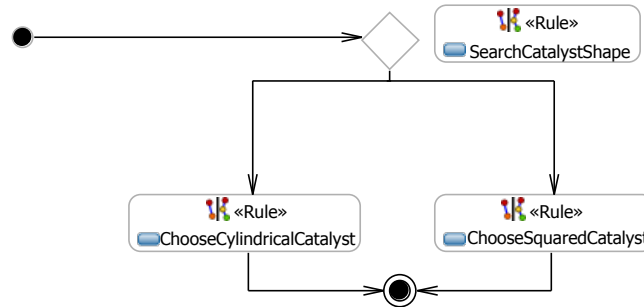


Abbildung 2.8: Beispiel eines Produktionssystems, welches im Entscheidungsknoten (Raute), auf Basis des in Regel *SearchCatalystShape* aus dem Entwurfsgraph gefilterten Katalysatorformparameters, die entsprechende Einbauregel aufruft.

Die Entscheidungsknoten erlauben damit, Entwurfsentscheidungen auf Basis des aktuellen Zustandes des Entwurfsgraphen zu modellieren und entsprechend spezifische Regelwerke je nach Entscheidungsausgang aufzurufen und auszuführen. Das UML-Aktivitätsdiagramm erlaubt somit eine konsistente Abbildung des Entwurfsprozesses inklusive der Modellierung von einzelnen, elementaren Entwurfsentscheidungen.

Im UML-Aktivitätsdiagramm gibt es sogenannte Java-Regeln als Werkzeug zur direkten Ausführung von Programmcodes. Damit lassen sich im Entwurfsprozess Algorithmen implementieren und direkt, auf der Ebene des universellen Datenmodelles, ausführen.

2.3.6. Lösungspfadgenerator

Der dargestellte wissensbasierte Entwurf mit Entwurfssprachen wird durch einen sogenannten Lösungspfadgenerator komplettiert. Dieser ist in dem verwendeten Entwurfscompiler integriert und dient in erster Linie zur automatisierten Auflösung und Berechnung von (nicht-linearen) Gleichungssystemen oder Systemen von Zwangsbedingungen im konzeptionellen Vorentwurf [171]. Damit erlaubt der Lösungspfadgenerator (siehe Kapitel 2.2.4) eine automatisierte, konsistente Bewertung eines Entwurfes.

Der in [171] dargestellte graphentheoretische Algorithmus zur Auflösung von Gleichungs- und Zwangsbedingungssystemen ist im verwendeten Entwurfscompiler 43V2 [90] integriert und um eine Anbindung an ein Computeralgebrasystem zur automatischen Berechnung von symbolischen Gleichungen ergänzt [159]. Dies erlaubt die vollautomatisierte Verarbeitung der im Entwurfsgraphen gespeicherten analytischen Ausdrücke und Zwangsbedingungen. Im Zusammenspiel mit den Prozessketten des Entwurfscompilers können damit Entwürfe, auf Basis des vom Lösungspfadgenerator verarbeiteten Gleichungs- beziehungsweise Zwangsbedingungssystems, in Abhängigkeit von Analyseergebnissen, zum Beispiel aus den Prozessketten, synthetisiert werden. In der Literatur sind ähnliche Systeme zur computergestützten Verarbeitung von Zwangsbedingungen im Entwurf bekannt [171, 86, 145]. Diese weisen aber nicht das hohe Maß an informationstechnischer Integration auf, welche mit der Verwendung generischer Entwurfssprachen mit einheitlichem Datenmodell erzielt werden kann.

Der Lösungspfadgenerator verarbeitet das Gleichungssystem, welches bei der inkrementellen, regelbasierten Erstellung des Entwurfsgraphen, aus den im Klassendiagramm der Entwurfssprache definierten Gleichungen entsteht. Er bringt dabei die Gleichungen, in Abhängigkeit der als gegeben deklarierten Variablen, in eine korrekte Lösungssequenz. Die symbolischen Gleichungen werden dann an das angeschlossene Computeralgebrasystem zur unmittelbaren numerischen Lösung beziehungsweise symbolischen Auflösung des Gleichungssystems nach den unbekanntenen Variablen übergeben.

Die Funktion des Lösungspfadgenerators soll im Folgenden näher erläutert werden: Ein Gleichungssystem (2.13) bestehe aus M Variablen $x_1, x_2, \dots, x_i, \dots, x_M$ und N Gleichungen $f_1, f_2, \dots, f_j, \dots, f_N$. Die Gleichungen wiederum geben Abhängigkeiten zwischen den Variablen wieder und sollen in ihrer Gesamtheit das folgende Gleichungssystem ergeben:

$$\begin{aligned} f_1 &= f_1(x_1, x_2, x_i) \\ f_2 &= f_j(x_2, x_3, x_M) \\ &\vdots \\ f_N &= f_N(x_3, x_5, x_k) \end{aligned} \tag{2.13}$$

In Abhängigkeit der Anzahl der gegebenen Variablen L ergeben sich überbestimmte $M - L < N$, unterbestimmte $M - L > N$ und (bestimmte) Systeme mit $M - L = N$. Letztere Bedingung stellt bei mathematisch wohlformulierten Gleichungssystemen im Allgemeinen eine notwendige Bedingung zum Auffinden von Lösungen dar. Für eine detailliertere Betrachtung sei auf die Arbeit von Bölling [28] verwiesen.

Der regelbasierte Aufbau des Entwurfsgraphen und des damit verbundenen Gleichungssystems garantiert bei entsprechender Formulierung der Regeln die Generierung bestimmter Gleichungssysteme. Damit müssen im Bereich der Entwurfssprachen im Allgemeinen keine Heuristiken zur Reparatur bzw. Modifikation von unter- oder überbestimmten Systemen eingesetzt werden [35]. Unter- oder Überbestimmtheiten weisen im Kontrast dazu eher auf ein (physikalisch) unvollständiges oder fehlerhaft modelliertes Modell hin. Dieses soll durch den durchgehend regelbasierten Aufbau des Entwurfes in den Entwurfssprachen, per Definition (Kapitel 2.3: "Jede in der Entwurfssprache mögliche beziehungsweise erlaubte Kombination der Vokabeln stellt einen gültigen Entwurf dar.") gerade verhindert werden.

Das Auffinden einer korrekten Lösungssequenz im Lösungspfadgenerator erlaubt aber noch keine Aussage über die mathematische Lösbarkeit des Zwangsbedingungs- / Gleichungssystems [28]. Beispielsweise kann es miteinander unverträgliche Gleichungen geben, welche keine mathematische Lösung des Gleichungssystems, trotz dem Auffinden einer korrekten Lösungssequenz, erlauben.

Die mathematisch wohlformulierten Gleichungen können mit einem Lösungspfadgenerator automatisiert nach beliebigen Variablen aufgelöst und berechnet werden. Das heißt, dass sich während des Prozesses der regelbasierten Modellgenerierung die Lösungssequenz durch Veränderungen der Deklaration der gegebenen Variablen im Gleichungssystem ändern kann. Dies entspricht einer Variation der im Entwurf fest vorgegebene Anforderungen (*Requirements*) und damit einer automatisierten Berechnung verschiedener Entwurfsszenarien im Sinne einer "Was-Wäre-Wenn"-Analyse auf Anforderungsebene. Der Lösungspfadgenerator kann dann, jeweils in Abhängigkeit der Topologie des Gleichungssystems sowie der als gegeben deklarierten Variablen, eine Lösungssequenz auffinden. Damit stellt der Lösungspfadgenerator ein deklaratives Verfahren zur Auflösung von Zwangsbedingungen und Abhängigkeiten im wissensbasierten Entwurf dar und ist damit als ein Werkzeug der *deklarativen Wissensverarbeitung* zu bezeichnen [156].

Der im Entwurfscompiler 43V2 [90] implementierte Lösungspfadgenerator basiert auf dem Algorithmus von Serrano [171]. Dieser stellt sich als Hintereinanderausführung verschiedener Graphenalgorithmen auf dem Constraint Netzwerk, das aus den Gleichungen und Variablen des Gleichungssystem gebildet wird, dar. Die Einzelschritte des Lösungspfadgenerators werden im Folgenden an dem in [155] von Rudolph gezeigten Beispiel des symbolischen Gleichungssystem einer Gasturbine (2.14) dargestellt. Hier soll das Gleichungssystem nur in Form der Zuordnung der Variablen zu den Funktionen wiedergegeben werden. Im Rahmen dieser Einführung soll nur die prinzipielle Funktion des Auffindens einer Lösungspfadsequenz durch den graphenalgorithmischen Lösungspfadgenerator gezeigt werden. Die ausführlichen, physikalisch motivierten Gleichungsausdrücke sind in [155] im Anhang A3 im Detail zu finden.

$$\begin{aligned}
f1 &= f1(x_1, x_2, x_3) \\
f2 &= f2(x_3, x_4, x_5, x_6, x_7) \\
f3 &= f3(x_4, x_5, x_6, x_8, x_9) \\
f4 &= f4(x_2, x_4, x_5, x_9, x_{10}) \\
f5 &= f5(x_1, x_8, x_{11}) \\
f6 &= f6(x_7, x_{12}, x_{13}, x_{14}) \\
f7 &= f7(x_6, x_7, x_{12}, x_{15}, x_{16}) \\
f8 &= f8(x_9, x_{10}, x_{16}, x_{17}, x_{18}) \\
f9 &= f9(x_{12}, x_{15}, x_{19}) \\
f_{10} &= f_{10}(x_{15}, x_{18}) \\
f_{11} &= f_{11}(x_{12}, x_{17})
\end{aligned} \tag{2.14}$$

Im ersten Schritt wird das Gleichungssystem in einen bipartiten Graphen G überführt (Abbildung 2.9). E sei dabei die Menge der Kanten zwischen den disjunkten Teilmengen der Variablen X und der Gleichungen F , welche die Zuordnung der Variablen zu den Funktionen repräsentieren. Der bipartite Graph $G(X, F, E)$ des Gleichungssystem wird damit aus den disjunkten Teilmengen der Gleichungen $F = f_1, \dots, f_i, \dots, f_N$ und Variablen $X = x_1, \dots, x_j, \dots, x_M$ gebildet.

Der nächste Schritt beinhaltet die Zuordnung jeweils einer unbekannt Variablen zu jeweils einer Gleichung, die später zur Bestimmung der zugeordneten Unbekannten dienen soll. Dazu wird ein Maximum Bipartite Graph Matching Algorithmus verwendet [87, 5, 124]. Wenn alle Variablen des Systems eindeutig den Gleichungen zugeordnet werden können, spricht man von einem perfekten Matching⁸ (Abbildung 2.10).

Darüber hinaus existieren Matching Algorithmen, welche innerhalb des Matchings das Gleichungssystem in über-, unter- und bestimmte Teilgleichungssysteme zerteilen können (Canonical Decomposition) [35, 53]. Die Bedingung $M - L = N$ stellt eine notwendige aber keine hinreichende Bedingung für die perfekte Zuordnung der unbekannt Variablen zu den Gleichungen dar [171]. Das System kann trotz Erfüllung der vorgenannten Bedingung in unter- und überbestimmte Teilsysteme zerfallen (siehe Abbildung 2.11).

Bei überbestimmten (Teil-)Systemen $M - L < N$ können ein oder mehrere maximale Matchings gefunden werden. Diese sind jedoch nicht perfekt, da es überzählige Gleichungen gibt. Eine mögliche Problemlösung wäre es, die überzähligen Gleichungen zunächst zu streichen und

⁸Ein Matching, dessen Kanten alle Knoten des Graphens enthalten, im betrachteten Fall alle Gleichungen und alle unbekannt Variablen, wird als perfektes Matching bezeichnet. Die bekannten Variablen werden beim Matching ignoriert.

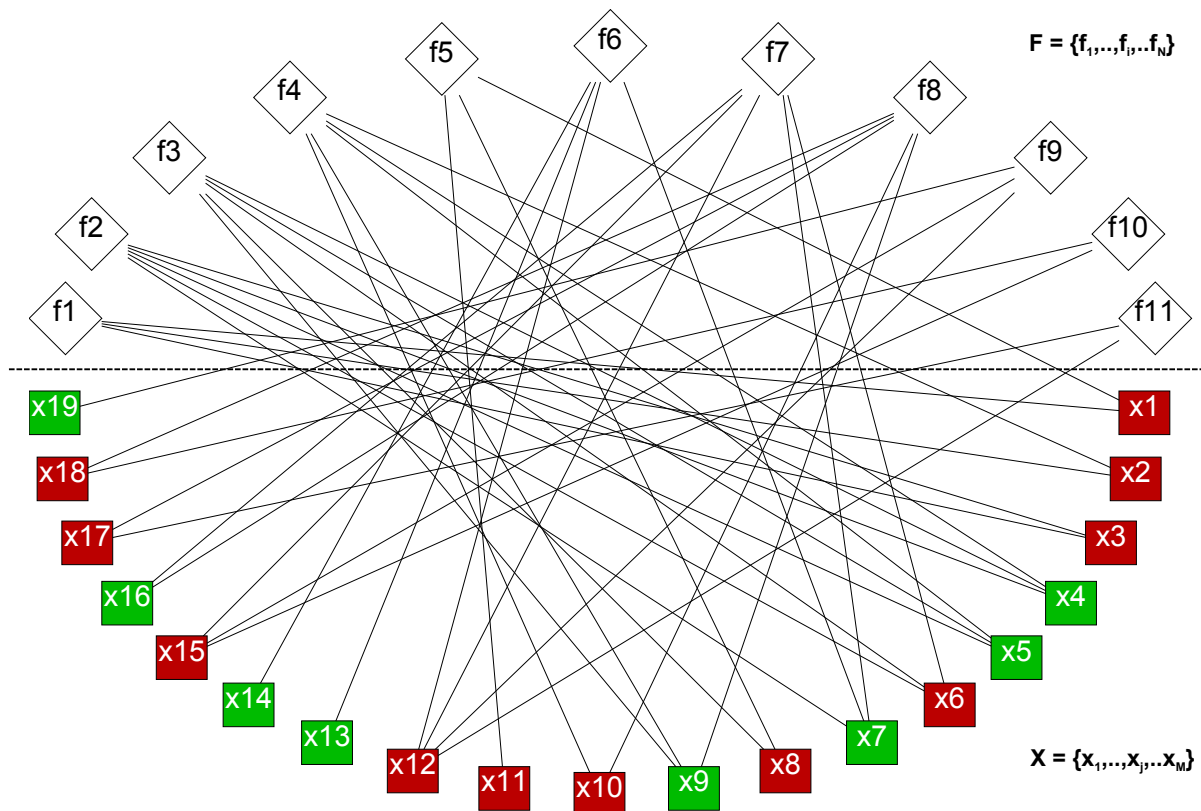


Abbildung 2.9: Bipartiter Graph der Zuordnung der Variablen zu den Gleichungen des Gleichungssystem: Gebildet aus den disjunkten Teilmengen F der Gleichungen (Rauten) und X der Variablen (Quadrate). Gegebene Variablen sind grün und gesuchte Variablen rot eingefärbt.

im Anschluss daran die überzähligen Gleichungen auf Redundanz bzw. Widerspruch mit der gematchten Gleichung zu überprüfen [171].

Bei unterbestimmten (Teil-)Systemen $M - L > N$ kann ebenfalls kein perfektes Matching gefunden werden. Nach dem Matching bleiben nicht zugeordnete Variablen stehen. Eine mögliche Konfliktlösung besteht darin, die Größe dieser Variablen zu schätzen oder zu raten [171]. Dieser Fall würde jedoch eine Unterbrechung des automatisierten Entwurfsablaufes bedeuten.

Nach der Zuordnung der unbekanntenen Variablen zu den Gleichungen wird nun ein gerichteter Graph der Abhängigkeiten erzeugt (Abbildung 2.12). Die gerichteten Kanten geben die Abhängigkeiten der Variablen untereinander wieder. Zeigt der Pfeil von Variable x_i auf Variable x_j so hängt die Variable x_i von der Variable x_j ab. Das heißt Variable x_i kann erst bestimmt werden, wenn Variable x_j bekannt ist.

Dieser Graph wird direkt aus dem Maximalen Matching Graph nach folgendem Schema bestimmt: Die Gleichungsknoten werden entfernt und die zuvor im Matching zugeordnete unbekanntene Variable dient als Ausgangsknoten der neuen, gerichteten Abhängigkeitskanten. Diese neuen Abhängigkeitskanten sollen dann auf die ungematchten, mit dem zu ersetzenden Gleichungsknoten verbundenen, Variablenknoten zeigen (vergleiche dazu die Abbildungen 2.10 und 2.12). Damit ergibt sich ein gerichteter Graph der Abhängigkeiten zwischen den Variablen.

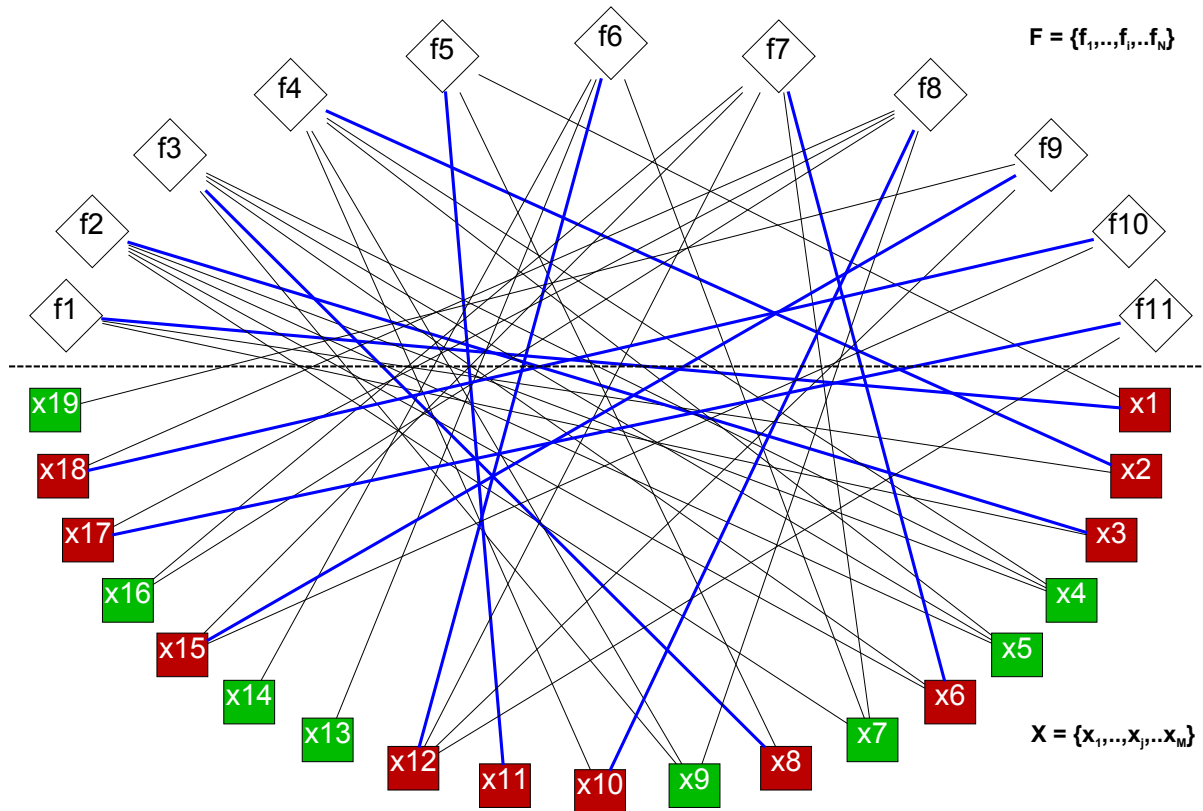


Abbildung 2.10: Perfektes, maximales Matching (blaue Kanten) zwischen den unbekanntem Variablen (rot) und den Gleichungen.

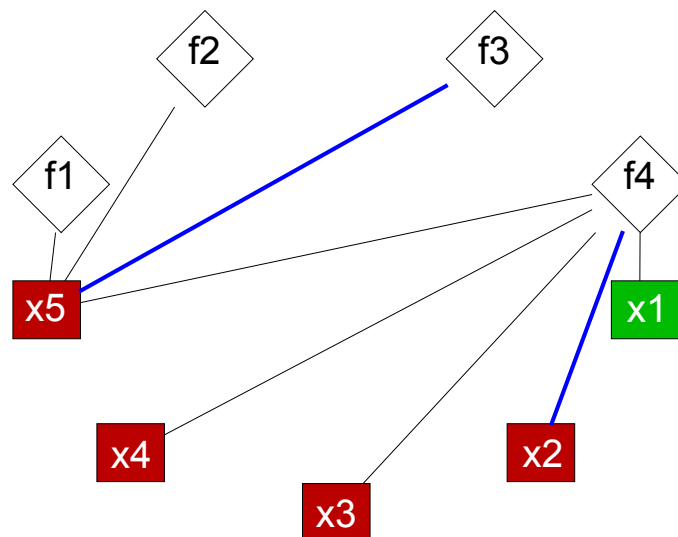


Abbildung 2.11: Maximales, aber nicht perfektes Matching von Gleichungen F und Variablen X mit $M - L = N$.

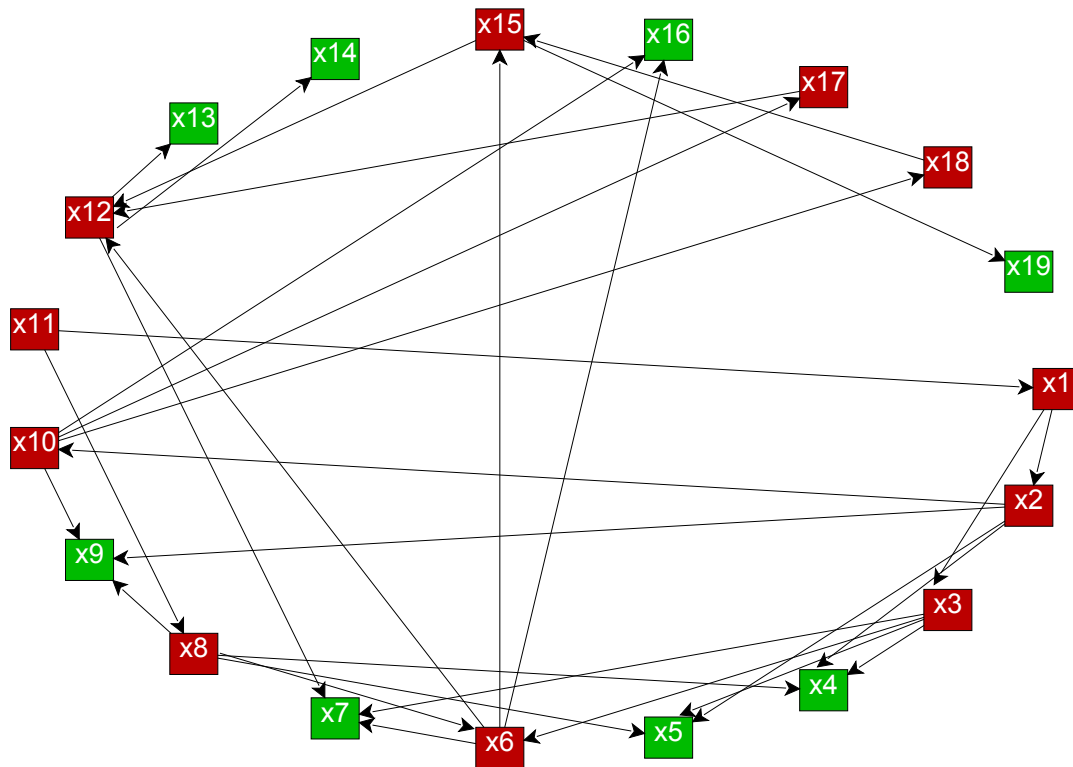


Abbildung 2.12: Gerichteter Graph der Abhängigkeiten der Variablen.

Der so gefundene gerichtete Graph muss nun auf zyklische Abhängigkeiten geprüft werden⁹ (u.a. [159]). Eine zyklische Abhängigkeit bedeutet, dass es Pfade entlang der gerichteten Kanten gibt, welche wieder auf einen bereits besuchten Knoten führen. Diese auch als Strongly Connected Components bezeichneten zyklischen Abhängigkeiten können mit entsprechenden Algorithmen gefunden werden [188, 124]. Treten zyklische Abhängigkeiten auf, so bedeutet dies für den Lösungspfad, dass die zyklisch voneinander abhängenden Variablen gleichzeitig, also innerhalb eines Schrittes der Lösungssequenz, gelöst werden müssen [171]. Eine Lösungssequenz kann dadurch gefunden werden, dass die Variablen eines Zyklusses in einen einzigen Superknoten zusammengefasst werden. Dabei werden die übrigen Abhängigkeiten beibehalten.

Im weiteren Fortgang wird der azyklische, gerichtete Graph der Abhängigkeiten so sortiert, dass sich die Lösungssequenz aus der geordneten Reihenfolge der unbekanntnen Variablen ergibt. Dazu werden zuerst die gegebenen Variablen, sowie die Kanten, welche auf die gegebenen Variablen zeigen, aus dem Graph gelöscht (Abbildung 2.13).

Der übrigbleibende Graph wird nun, beginnend mit einem Variablenknoten von dem keine andere Variable abhängt, invers topologisch sortiert [99]: Die Variablen repräsentierenden Knoten werden so entlang einer gegebenen Richtung angeordnet, dass die gerichteten Kanten alle entgegengesetzt der vorgegebenen Richtung zu liegen kommen (*inverse* topologische Sortierung).

Abbildung 2.14 zeigt eine solche topologische Sortierung. Die Sortierungsrichtung ist dabei vertikal, von unten nach oben. Die unterste Variable stellt die zuerst zu lösende Variable dar. Im dargestellten Graph können die jeweils auf einer horizontalen Linie liegenden Variablen simultan, unter Verwendung der im Matching jeweils zugeordneten Gleichungen, gelöst werden.

⁹Die zyklischen Abhängigkeiten haben sich in der Praxis als selten erscheinende Spezialfälle in den zahlreichen Beispielanwendungen herausgestellt.

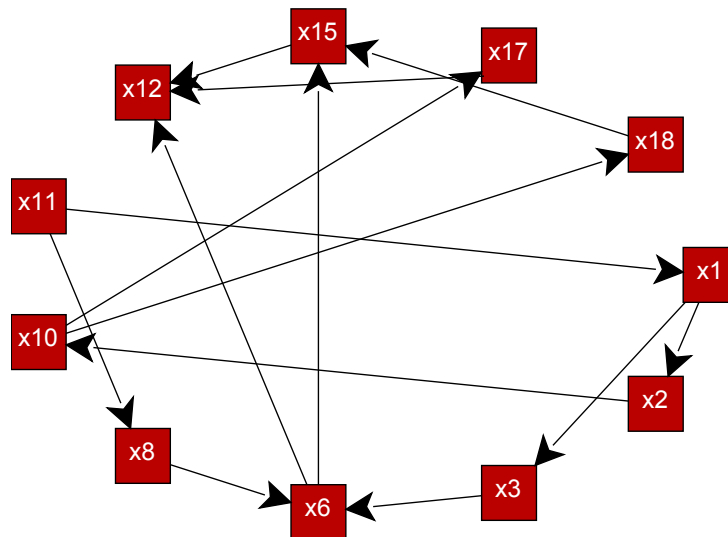


Abbildung 2.13: Gerichteter Graph der Abhängigkeiten nach der Elimination der Knoten der gegebenen Variablen.

Performanz Tabelle 1 zeigt abschließend typische Aufwände für die oben dargestellten Graphenalgorithmus. In der Aufwandsnotation gibt V die Gesamtzahl der Knoten an und E steht für die Anzahl der Kanten. Die dargestellten Aufwände dienen im weiteren Verlauf als Vergleichsgrößen zur Bewertung des selbstassemblierenden Lösungspfadgenerators.

Aufgabe	Algorithmus	Aufwand	Quelle
Maximum Bipartite Matching	Hopcraft-Carp	$O(\sqrt{V}E)$	[87]
Maximum Bipartite Matching	Alt-Blum-Mehlhorn-Paul	$O(V^{1.5}\sqrt{E/\log V})$	[5]
Finden zyklischer Abhängigkeiten	Tarjan	$O(V + E)$	[188]
Topologische Sortierung	Kahn	$O(V + E)$	[99]

Tabelle 1: Aufwände von Graphenalgorithmus, die in Implementierungen des graphenalgorithmischen Lösungspfadgenerators eingesetzt werden.

2.4. Computergestützte Pfadsuche und Verrohrung

In diesem Kapitel werden Algorithmen zur computergestützten, automatisierten Suche von Pfaden und Rohrleitungen dargestellt. Im ersten Schritt sollen Algorithmen zur Pfadsuche auf Graphen dargestellt werden. Im zweiten Schritt werden ausgewählte repräsentative Anwendungen dieser Algorithmen im relevanten Themenfeld aufgezeigt.

2.4.1. Algorithmen

Pfadsuchalgorithmen werden eingesetzt um Verbindungen zwischen einem Start- und einem Zielknoten auf einem Graph zu finden. Das klassische Pfadsuchproblem stellt das Auffinden eines kürzesten Pfades auf einem gewichteten Graphen dar und ist in der Graphentheorie unter der Bezeichnung Shortest Path Problem bekannt [190, 198].

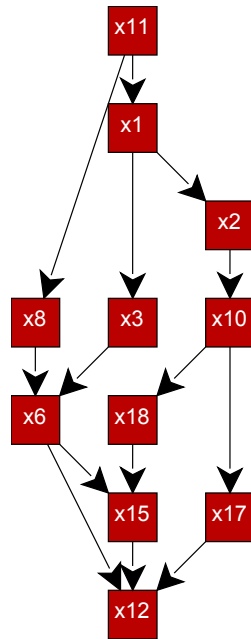


Abbildung 2.14: Topologisch sortierte, unbekannte Variablen. Sequenz der Abarbeitung von unten nach oben, Abhängigkeiten von oben nach unten.

In diesem Kapitel soll ein kurzer Überblick über den Stand der Literatur der Pfadsuchalgorithmen vermittelt werden. Die Darstellung kann auf Grund des Umfangs des Themengebietes keinen Anspruch auf Vollständigkeit erheben, sondern soll sich auf die kurze Wiedergabe der wesentlichen Probleme und Lösungsansätze beschränken¹⁰.

Betrachtet man die Historie der Pfadsuchalgorithmen, so sind die ersten Algorithmen im Bereich der Graphentheorie zu finden. Der Pfadsuchalgorithmus von Dijkstra [52] findet in einem mit nicht-negativen Gewichten versehenen Graph jeweils die kürzesten Pfade von einem Startknoten zu allen anderen Knoten des Graphen. Der Bellman-Ford Algorithmus [18] überwindet, auf Kosten eines höheren Aufwandes, die Beschränkung auf Graphen mit nicht-negativen Gewichten. Dagegen findet der Floyd-Warshall-Algorithmus [63] innerhalb eines Durchganges den minimalen Abstand (Summe der Gewichte) aller Knotenpaare des Graphes, findet dabei aber nicht die Knotenliste des jeweiligen Pfades. Der A*-Algorithmus [79] erweitert den Dijkstra-Algorithmus um eine Suchheuristik und erzielt so bessere Laufzeiten. Konkrete Implementierungen solcher Graphenalgorithmen sind in entsprechenden Standardwerken zu finden [43].

Breite Anwendung finden Pfadsuchalgorithmen bis heute im Chipdesign. Zur Planung von hoch-integrierten Schaltkreisen (sogenannten Very-Large-Scale Integration (VLSI)-Chips). Die für die Integration einer immer größeren Anzahl von Transistoren notwendige Automatisierung im Chipdesign beinhaltet unter anderem die computergestützte Suche nach einer optimalen Führung der Leiterbahnen. Diese Pfadsuche findet in der Regel auf einem äquidistanten Gitter statt (Manhattan-Metrik). Ein erster Pfadsuchalgorithmus für das Routing der Leiterbahnen in einer Ebene stellt der Lee-Maze-Algorithmus [118] dar. Dieser wurde in [179] so erweitert, dass eine parallele Suche verschiedener Leiterbahnen ermöglicht wurde. Mit der zunehmenden Integration wurden mehrschichtige Chipdesigns und in Folge entsprechend angepasste Pfad-

¹⁰In der vorliegenden Arbeit werden zwar Pfadsuchalgorithmen verwendet sowie ein Ansatz für den computergestützten Entwurf von Verrohrungen mit vorgegebenen Rohrbauteilen ausgearbeitet, es findet aber keine explizite (Weiter-)Entwicklung eines solchen Pfadsuchalgorithmus statt.

suchalgorithmen notwendig [41, 42]. Andere Ansätze schließen auch die Suche nach diagonalen Routen ein [109].

Im Bereich der Computernetzwerke kommen Pfadsuchalgorithmen für die Festlegung der Routingstrecken der Datenpakete zur Anwendung. Dort stellt sich das dynamische beziehungsweise adaptive Routing (Netzwerkpfade werden im laufenden Netzbetrieb geändert) in sich verändernden Netzwerktopologien (An- und Abmeldung von Netzwerkknoten) als Hauptproblem dar [10, 122]. Für den Entwurf der Verrohrung von SCR-Systemen im Rahmen dieser Arbeit spielt das dynamische Routing aber keine Rolle und sei nur der Vollständigkeit halber erwähnt.

2.4.2. Anwendungen

Im Bereich des Ingenieurwesens finden sich verschiedene Anwendungen von Pfadsuchalgorithmen. Eine breite Anwendung stellt die Pfad- und Bewegungsplanung von Robotern und anderen sich autonom bewegenden Systemen dar [95, 116]. Der Forschungsbericht [62] schildert im Bereich der Bewegungsplanung von Robotern einen Algorithmus zum Auffindens eines Pfades, welcher minimale Biegungen aufweist und somit auch prinzipiell für das Auffinden von Rohrpfaden interessant ist. In [186] ist im Bereich der automatisierten Schiffsroutenplanung ebenfalls ein Algorithmus vorgeschlagen, der Pfade unter Minimierung der Biegung finden kann.

Ein weiteres großes Anwendungsgebiet stellt der automatisierte Entwurf von Verrohrungen dar. Im Bereich der Luftfahrttechnik werden oben beschriebene Pfadsuchalgorithmen im Rahmen wissenschaftlicher Ingenieursoftware für den Entwurf der Kabel- und Rohrführung innerhalb der Flugzeughülle verwendet [194]. Der Einsatz der automatisierten Pfadsuche im Entwurf der Rohrleitungs-ausrüstung von verfahrenstechnischen und industriellen Anlagen [96, 75] sowie der automatisierte Entwurf von technischer Gebäudeausrüstung im Architekturwesen [123] ist ebenfalls Gegenstand der Forschung.

Im Bereich der Verrohrung von Maschinenräumen sind zahlreiche Automatisierungsansätze zu finden: Diese reichen von Expertensystemen [100, 139] über Pfadsuchsysteme, die mit einem genetischen Algorithmus zur Optimierung gekoppelt sind [96, 91], bis hin zu nicht-deterministischen Optimierungsmethoden [11] (z.B. Partikel-Schwarm-Optimierung). Daneben existieren integrierte Systeme, die eine simultane Optimierung von Verrohrungspfaden und Schiffsrumpfstruktur erlauben [152]. Die simultane Variation der Schiffsrumpfstruktur gibt implizit den Bauraum für die Verrohrungssuche vor.

Die bisher genannten Anwendungen haben gemeinsam, dass sich die verwendeten Pfadsuchalgorithmen und Techniken im Wesentlichen auf den Entwurf von rechtwinkligen und achsenparallelen Rohrführungen beschränken. Die Arbeit [7] bildet dabei eine Ausnahme. Sie beschreibt die automatisierte Verrohrung um Hindernisse unter Verwendung von nicht-rechtwinkligen Rohrbögen. Die nicht-rechtwinkligen Rohrbögen werden aber nur paarweise, in einer Ebene liegend, zum Ausweichen vor Hindernissen verwendet. Damit bleibt die Einschränkung auf achsenparallele, gerade Rohrstrecken erhalten und es können keine schief durch den Bauraum laufenden Rohrstrecken gefunden werden.

Ein alternativer Ansatz zur Synthese nicht-rechtwinkliger Rohrrouten ist in [187] am Beispiel der Verrohrung einer Industrieanlage dargestellt. Darin wird ein Pfad nach dem Konzept des Shape Annealings [37] synthetisiert, welches eine Verbindung einer Gestaltsgrammatik (Shape Grammars) [8] mit einem stochastischen Optimierungsverfahren (Simulated Annealing) [103] darstellt. Zwangsbedingungen, wie zum Beispiel bevorzugte Bogenwinkel, Abstände zu Objek-

ten beziehungsweise Verletzungen der Bauraumgrenzen, werden über eine entsprechend gestaltete Bewertungsfunktion berücksichtigt.

Die Shape Grammars bestehen aus Regeln zum Aufbau und zur Manipulation eines Pfades. Dies ist mit dem oben beschriebenen, aber allgemeiner gefassten Entwurfssprachenparadigma vergleichbar, welches zu jedem Zeitpunkt der Regelausführung, zum Auf- oder Umbau eines Entwurfsobjektes, die formale Korrektheit des Objektes selbst, im betrachteten Fall der des Pfades, gewährleistet.

Das Simulated Annealing Schema (siehe auch Kapitel 2.5.2) führt die Auf- und Umbauregeln auf Basis der Bewertung des jeweils aktuellen Pfades, unter Einbeziehung einer stochastischen Unschärfe, aus und führt so zu einer Gestaltoptimierung des Pfades. Die Anwendung in [187] beschränkt sich bezüglich der Optimierung auf die gleichzeitige Verhinderung der Durchdringung von Hindernissen sowie auf die Minimierung der Pfadlänge und der Gesamtbiegung des Pfades. Geometrische Zwangsbedingungen, wie sie beispielsweise bei einer Verrohrung unter Verwendung von Standardrohrelementen vorkommen, werden dabei nicht berücksichtigt. Der Grundgedanke des Shape Annealings wird im weiteren Verlauf der Arbeit in Form der Verrohrungssprache aufgegriffen.

2.5. Komplexe Systeme

Dieses Kapitel stellt für die vorliegende Arbeit relevante Aspekte und Methoden der Behandlung komplexer Systeme dar: Anfangs wird das charakteristische Verhalten nichtlinearer Systeme erläutert. In den weiteren Abschnitten werden der Optimierungsalgorithmus der simulierten Abkühlung sowie mathematische Dimensionsbegriffe näher dargestellt. Der Optimierungsalgorithmus wird im weiteren Verlauf im Rahmen der Verrohrungssprache zur Synthetisierung von Rohrstrecken eingesetzt. Die Analyse der mathematischen Dimensionen innerhalb des Entwurfes wird für die Ableitung einer Entwurfssequenz verwendet.

2.5.1. Nichtlineares System

Lineare Systeme weisen die Eigenschaft auf, dass eine Linearkombination von Eingangssignalen x_i äquivalent zu einer Linearkombination der jeweiligen Ausgangssignale $y = f(x_i)$ ist [65]:

$$y = L(a_1 \cdot x_1 + a_2 \cdot x_2) = a_1 \cdot L(x_1) + a_2 \cdot L(x_2) \quad (2.15)$$

Operatoren $L(x)$, welche Gleichung (2.15) erfüllen, werden als lineare Operatoren bezeichnet. Als nichtlinear werden Systeme bezeichnet, welche diese Eigenschaft nicht erfüllen. Ein einfaches Beispiel für ein nichtlineares System stellt zum Beispiel die Heavyside-Funktion dar:

$$y = f(x) = H(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases} \quad (2.16)$$

Das Verhalten solcher nichtlinearer Systeme kann empfindlich von (der Genauigkeit) von Anfangswerten abhängen. Dies wird in der Systemtheorie auch als Butterfly-Effekt bezeichnet [119]: Geringste Abweichungen in den Eingangsgrößen können zu einem vollständig unterschiedlichem Systemverhalten führen.

Die Strömung im Abgassystem und insbesondere das Phänomen der Turbulenz in der Strömung des Abgassystems stellt ein klassisches nichtlineares System dar: Die Strömungsdynamik im

Abgassystem wird durch ein System nichtlinearer Gleichungen, den sogenannten Navier-Stokes-Gleichungen, beschrieben [50]. Beispielsweise hängt die Wirbelablösung in der Abgasströmung, welche für die Effizienz der Abgasreinigung relevant sein kann, empfindlich von Anfangs- und Randbedingungen sowie der Diskretisierung der Geometrie ab [50]. Schon kleine Änderungen an der Geometrie oder der Diskretisierung der Geometrie können so zu großen Änderungen des Ergebnisses der Strömungssimulation führen.

Solch nichtlineares Verhalten des zu entwerfenden Produktes und der am Entwurf beteiligten Systeme muss in der Entwicklung von Entwurfssprachen berücksichtigt werden. Die Abhängigkeit des Systemverhaltens von kleinsten Änderungen des Entwurfes impliziert im Allgemeinen die Notwendigkeit einer vollständigen Wiederausführung des Entwurfsprozesses, nach der Anwendung kleinster Änderungen der Anforderungen oder Randbedingungen.

2.5.2. Simulierte Abkühlung

Die Simulierte Abkühlung (Simulated Annealing) stellt eine stochastische Heuristik zur kombinatorischen Optimierung dar [103]. Dabei wird eine Gütefunktion $W(\vec{x})$ einer Konfiguration \vec{x} des zu optimierenden Systems definiert. Diese soll in der Optimierung entweder minimiert oder maximiert werden [103]. Das Vorgehen der simulierten Abkühlung lehnt sich an den Abkühlungs- und Erstarrungsprozess von Festkörpern an: Die physikalischen Zustände der Festkörper nähern sich bei einer (ausreichend langsamen) Absenkung der Temperatur immer mehr dem energetischen Grundzustand des Systems und damit dem Minimum der Energiefunktion an. Dieses Prinzip dient als Vorlage der simulierten Abkühlung zur Annäherung an das globale Minimum der Gütefunktion [103].

In der statistischen Physik ergibt sich die Wahrscheinlichkeit einen Zustand \vec{x} mit Energie $E(\vec{x}) \equiv W(\vec{x})$ bei einer Temperatur T anzutreffen proportional zum Boltzmann-Faktor [131]:

$$p(\vec{x}) \propto e^{-\frac{E(\vec{x})}{k_b \cdot T}}, \quad (2.17)$$

mit der Boltzmann-Konstante k_b .

In der Optimierung mit Simulierter Abkühlung wird der Systemzustand iterativ verändert:

$$\vec{x}_{neu} = \vec{x}_{alt} + \Delta\vec{x}. \quad (2.18)$$

Nach [103] entspricht dies, in der Betrachtungsweise der statistischen Physik, einer mikroskopischen Zustandsänderung des zugrundeliegenden physikalischen Systems.

Die neuen Zustände \vec{x}_{neu} einer iterativen Zustandsänderung werden im Optimierungsschema der simulierten Abkühlung nur dann akzeptiert, wenn sie folgende Kriterien erfüllen¹¹ [103]:

1. Eine bessere Güte als der Vorgängerzustand \vec{x}_{alt} , $W(\vec{x}_{neu}) < W(\vec{x}_{alt})$ aufweisen, oder
2. die Verschlechterung der Güte $\Delta W = \vec{x}_{neu} - W(\vec{x}_{alt}) > 0$ mit einer Wahrscheinlichkeit von $p = e^{-\frac{\Delta W}{T}}$ akzeptiert wurde (Temperaturparameter T mit der Dimension [W]).

Die zweite Bedingung entspricht dem Metropolis-Algorithmus [125] und stellt sicher, dass für höhere Temperaturparameter T der Zustandsraum weiträumig abgesucht wird und das System nicht in einem lokalen Minima stecken bleibt [103]. Nach [103] entspricht dies, im Bilde des physikalischen Systems, einem Kontakt mit einem Wärmebad der Temperatur T . Die

¹¹Die Kriterien basieren auf der Annahme, dass die Gütefunktion minimiert werden soll. Eine Maximierung der Gütefunktion kann beispielsweise über die Kehrwertbildung der Gütefunktion auf das Minimierungsproblem zurückgeführt werden

Zustandsdichtefunktion des physikalischen Systems nähert sich dann in der Optimierung der Boltzmannverteilung an.

Der Optimierungsprozess der simulierten Abkühlung wird nun durch eine sukzessive Reduktion des Temperaturparameters T vollzogen. Nachdem eine stationäre Verteilung der Zustände für ein Temperaturniveau erreicht wurde, wird der Temperaturparameter um einen gegebenen Wert oder Faktor erniedrigt. Dieser Ablauf wird als Abkühlungsplan (Annealing Schedule) bezeichnet [103]. Mit abnehmender Temperatur werden zunehmend, entsprechend der oben definierten Kriterien, nur noch Zustände akzeptiert, die eine Verbesserung oder eine minimale Verschlechterung der Güte W bewirken [103].

Für die Bewertung der Stationarität der Zustandsverteilung bei einer festen Temperatur T sowie zur Berechnung eines Temperaturdekrementes ΔT gibt es verschiedene Vorschläge in der Literatur: Die Quelle [115] gibt einen Überblick über verschiedene Abkühlungspläne, welche Methoden zur Überprüfung der Stationarität eines Temperaturniveaus beinhalten. Die Arbeit [76] stellt notwendige Bedingungen an den Abkühlungsplan dar, die für das garantierte Auffinden eines globalen Optimums erfüllt werden müssen. Diese technischen Aspekte der simulierten Abkühlung sollen hier aber nicht näher betrachtet werden, da in der vorliegenden Arbeit ein vereinfachter Abkühlungsplan verwendet wird.

2.5.3. Dimensionsbegriffe

Dieser Abschnitt soll einen kurzen Überblick über den Dimensionsbegriff der Mathematik verschaffen. Dieser wird im Verlauf der Arbeit als Werkzeug zur Charakterisierung der Entwürfe im Entwurfsraum verwendet. Die Darstellung soll in kompakter Form die Vielfalt der mathematischen Dimensionsbegriffe zeigen und grundlegende, anschauliche Definitionen dieser Dimensionsbegriffe aufzeigen.

Eine mathematische Dimension kann sowohl konkreten als auch abstrakten Objekten zugeordnet werden und stellt ein Maß für die Komplexität des Objektes dar. Die im Allgemeinen geläufigste Anwendung des Dimensionsbegriffs findet sich in der Charakterisierung geometrischer Objekte. Dabei gibt die mathematische Dimension die Anzahl der (linear unabhängigen) Raumrichtungen an, welche das geometrische Objekt ausfüllt:

- Ein Punkt ist nulldimensional.
- Eine Linie ist eindimensional.
- Eine Fläche ist zweidimensional.
- Ein Volumen ist dreidimensional.

Entsprechend dem induktiven Dimensionsbegriff, welcher postuliert, dass der Rand eines Objektes der Dimension d die Dimension $d-1$ hat, wird der leeren Menge die Dimension $\dim(\emptyset) = -1$ zugeordnet [82]. Entsprechend lässt sich induktiv die Dimension eines Objektes über die rekursive Betrachtung der Ränder der jeweiligen niederdimensionaleren Objekte bestimmen: Ein Punkt wird von der leeren Menge berandet und hat entsprechend eine Dimension von $d = 1$. Eine Linie wird an ihren Enden von Punkten berandet und besitzt die Dimension $d = 2$, usw.

Die Lebesgue-Überdeckungsdimension (auch Topologische Dimension) wird mit dem Operator $\dim(X)$ eines (topologischen) Raumes oder Objektes X ausgedrückt. Die offene Überdeckung U von X setzt sich aus offenen Untermengen $U_i \in X$ mit $i \in I$ zusammen, deren Vereinigung wiederum X ergibt. Die Verfeinerung V der offenen Überdeckung U ist selbst wieder eine offene

Überdeckung. Sie ist dadurch charakterisiert, dass ihre Untermengen V_k mit $k \in K$ jeweils in einer der Untermengen U_i der Überdeckung U mit enthalten sind: $V_k \in U_i$.

Jedes Element von X solle in maximal $n+1$ der die Überdeckung bildenden Teilmengen vorkommen. Dann stellt $n+1$ die Ordnung der Überdeckung beziehungsweise die Höchstzahl der sich überschneidenden Teilmengen der Überdeckung dar. Die Lebesgue-Überdeckungsdimension ist nun so definiert, dass eine soweit verfeinerte Überdeckung existiert, welche eine minimale Ordnung $n+1$ aufweist, die dem Objekt X die Dimension n zuordnet. Dies ist in Abbildung 2.15 am Beispiel einer Linie mit Dimension $n=1$ dargestellt.

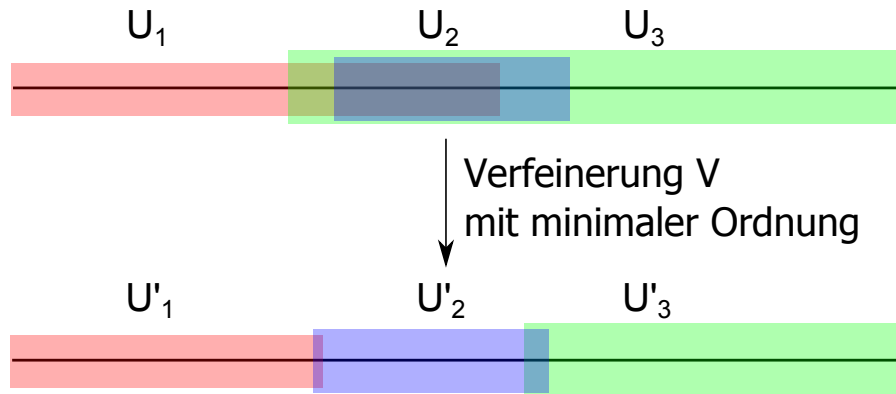


Abbildung 2.15: Bestimmung der Lebesgue-Überdeckungsdimension: Verfeinerung der Überdeckung einer Linie mit $n=1$, die minimale Ordnung aufweist.

Eine wichtige Eigenschaft der Lebesgue-Überdeckungsdimension n ist, dass diese mit der Dimension des euklidischen Raumes \mathbb{R}^n (Raum der Anschauung) übereinstimmt. Dieser ist in der Definition von [82] ein Vektorraum, der alle n -Tupel reeller Zahlen \mathbb{R} enthält und jeden Punkt P des Raumes den Ortsvektor \vec{OP} zwischen Ursprung O und P zuordnet.

In der linearen Algebra wird die Dimension eines Vektorraumes als die maximale Mächtigkeit einer Teilmenge linear unabhängiger Vektoren des Vektorraumes definiert. Dies entspricht der Anzahl der Vektoren in der Basis des Vektorraumes [82].

Die Hausdorff-Dimension erlaubt es metrischen Räumen¹² beziehungsweise Objekten in diesen Räumen eine Dimension zu zuweisen [82]. Der Wert der Dimension kann auch rational oder irrational sein und so Fraktale, die nicht-ganzzahlige Dimensionswerte besitzen, charakterisieren. Eine wichtige Eigenschaft der Hausdorff-Dimension ist, dass ihr Wert für geometrische Objekte mit ganzzahligen Dimensionswerten mit dem Wert des oben dargestellten topologischen Dimensionsbegriffes übereinstimmt.

Die Hausdorff-Dimension lässt sich anschaulich über ein Skalierungsgesetz skizzieren. Dazu wird die zur Überdeckung des Objektes notwendige Mindestzahl an (Hyper-)Kugeln, welche die Dimension des Raumes besitzen in dem das Objekt eingebettet ist, bestimmt. Diese Zahl der Kugeln sei $N(r)$ für einen gegebenen Kugelradius r . Im Grenzübergang des Kugelradius $r \rightarrow 0$ ergibt sich die Hausdorff-Dimension zu:

$$d_H = \lim_{r \rightarrow 0} \frac{\ln(N(r))}{\ln(\frac{1}{r})}. \quad (2.19)$$

Formal wird die Hausdorff-Dimension über das Hausdorff-Mass abgeleitet und berechnet [59], dies soll aber im Rahmen dieser Arbeit nicht näher dargestellt werden. Fraktale haben des Wei-

¹²Ein Metrischer Raum stellt eine Menge dar, zwischen deren Elemente ein Abstandsmaß definiert ist [82].

teren die Eigenschaft, dass sie Selbstähnlichkeit charakteristischer Strukturen über verschiedene Längen- bzw. Beobachtungsskalen aufweisen.

Auf der Hausdorff-Dimension aufbauend lässt sich ein weiterer anschaulicher Dimensionsbegriff definieren: Ein Objekt bestehe aus N verkleinerten Kopien seiner selbst, welche die Größe r aufweisen. Jede dieser verkleinerten Kopien bestehe wiederum aus N verkleinerter Kopien ihrer selbst. Dieser Rekursion wird die Ähnlichkeitsdimension in Gleichung (2.20) zugeordnet [82]:

$$d_{\text{Fraktal}} = -\frac{\log(N(r))}{\log(r)}. \quad (2.20)$$

Für einen Würfel, der selbst aus acht Würfeln halber Kantenlänge besteht, ergibt sich damit eine Ähnlichkeitsdimension von drei. Dies entspricht dem anschaulich zu erwartenden Wert.

Die Dimension eines Objektes kann näherungsweise bestimmt werden, in dem die Struktur des Objektes quantitativ $N(r)$ auf verschiedenen Beobachtungsskalen r charakterisiert (z.B. die Anzahl der selbstähnlichen Elemente) wird und $N(r)$ einem doppellogarithmischen Schaubild gegen verschiedene Beobachtungsskalen r aufgetragen wird [59]. Die aufgetragenen Werte kommen dann auf einer Geraden zu liegen, deren Steigung dem Dimensionswert analog zu Gleichung (2.20) entspricht.

Die Abbildung 2.16 zeigt eine in der Natur vorkommende fraktale Struktur eines Romanesco Blumenkohls. Der Skalenbereich des fraktalen Verhaltens ist in diesem realen Beispiel beschränkt. Dagegen gibt es synthetische Beispiele aus der Mathematik, wie die Cantor-Menge oder die Koch-Kurve, welche fraktales Verhalten auf beliebigen Längenskalen aufweisen [59].



Abbildung 2.16: Romanesco Blumenkohl weist, als in der Natur vorkommendes Objekt, selbstähnliche fraktale Strukturen auf einem beschränkten Skalen- beziehungsweise Längenbereich auf. Quelle: Aurelien Guichard, www.flickr.com.

Die Dimensionsdefinitionen aus den Gleichungen (2.20) und (2.19) haben einen sehr universellen Charakter. Sie benötigen zur Bestimmung letztendlich nur die Definition einer skalenabhängigen Quantität der Struktur des zu charakterisierenden Objektes und eine entsprechende korrespondierende Beobachtungsskala.

In der praktischen Anwendung sind diese Definitionen aber im Allgemeinen nicht ohne weitere Schwierigkeiten anwendbar. Eine numerisch einfacher zu berechnende Größe stellt die Box-Counting-Dimension dar. Sie basiert ebenfalls auf einer Berechnung der Skalierungseigenschaften des zu betrachtenden Objektes.

Zur Berechnung wird zunächst der Raum, in den das Objekt eingebettet ist, in Zellen mit Kantenlänge ϵ eingeteilt. Dann wird die Anzahl der Zellen $N(\epsilon)$, die von dem Objekt geschnitten werden, bestimmt. Im Grenzübergang verfeinerter Zellen, $\epsilon \rightarrow 0$, ergibt sich die Box-Counting-Dimension zu [59]:

$$d_{BC} = \lim_{\epsilon \rightarrow 0} \frac{\ln(N(\epsilon))}{\ln(\frac{1}{\epsilon})} \quad (2.21)$$

Für die numerische Berechnung kann der einbettende Raum in Zellen mit jeweils verschiedenen Zellgrößen ϵ eingeteilt werden. Für verschiedene Werte ϵ wird die Zahl $N(\epsilon)$ der vom Objekt geschnittenen Zellen gegen die Beobachtungsskala ϵ entsprechend Gleichung (2.21) doppellogarithmisch aufgetragen. Die Werte werden in der doppellogarithmischen Auftragung linear extrapoliert und die Steigung der so bestimmten Geraden ergibt eine Schätzung für die Dimension des Objektes [59]. Im Allgemeinen stimmen die Werte der verschiedenen Dimensionen für Fraktale nicht exakt überein, da die verschiedenen Dimensionsbegriffe unterschiedliche mathematische Eigenschaften aufweisen [59].

Abschließend sei noch die Rényi-Dimension (auch generalisierte Dimension oder q -Dimension) [147] dargestellt. Diese weist nach [149] Vorteile in der Dimensionsbestimmung auf, wenn nur eine beschränkte Datenbasis zur Verfügung steht. Dabei ist $\mu(B_i)$ die Dichte des Objektes (zum Beispiel Anzahl der Punkte) oder die Dichte einer Wahrscheinlichkeitsverteilung in einer Box B_i mit Kantenlänge ϵ :

$$D_q = \lim_{\epsilon \rightarrow 0} \frac{\ln(\sum_i \mu(B_i)^q)}{(1-q) \cdot \ln(\frac{1}{\epsilon})} \quad (2.22)$$

Der Parameter q kann verschiedene Werte annehmen. Für $q = 0$ ergibt sich eine Dimensionsdefinition analog zur Box-Counting-Dimension. Die Dimension mit $q = 1$ (unter Verwendung der Regel von L'Hospital) wird als Informationsdimension bezeichnet. Diese verallgemeinerte Dimensionen können beispielsweise in der Charakterisierung von Multifraktalen verwendet werden [178], deren Charakterisierung mehr als nur einen Dimensionswert (zum Beispiel ortsabhängige Dimensionswerte) benötigt.

2.5.4. Ordnungsmechanismen

Dieser Abschnitt soll eine kurze Definition der in der Arbeit verwendeten Begrifflichkeiten in der Behandlung ausgewählter Ordnungsmechanismen und Ordnungsphänomene umfassen und diese oft unscharf verwendeten Begriffe voneinander abgrenzen. Der Begriff der Ordnungsphänomene kann im Allgemeinen als Oberbegriff für sämtliche in (natürlichen) Systemen auftretenden Mechanismen und Phänomene der Herausbildung von Struktur verwendet werden.

Emergenz bezeichnet das Auftreten makroskopischer Artefakte, Strukturen und Verhaltens aus der mikroskopischen Interaktion der das makroskopische System bildenden Teilsysteme. Dabei sind die auftretenden makroskopischen Strukturen nicht explizit auf der mikroskopischen Ebene angelegt [206]. Die Emergenz als spezieller Ordnungsmechanismus steht wiederum in

Beziehung mit anderen Ordnungsphänomenen, wie der Selbstorganisation. Nach [206] werden vor allem die Definitionen der Begrifflichkeiten "Selbstorganisation" und "Emergenz" in der Literatur missverständlich verwendet.

Hinzu kommt der Begriff "Selbstassemblierung", der in der Wissenschaft verwendet wird um komplexe Systeme zu bezeichnen, welche aus einer zuvor bestehenden Unordnung durch innere Interaktionen Strukturen und Ordnung entstehen lassen [70]. Neben dem Begriff der "Selbstassemblierung" wird in der wissenschaftlichen Diskussion auch der bereits erwähnte Begriff der "Selbstorganisation" in der Beschreibung sich ordnender und strukturbildender Systeme verwendet. Beide Begriffe werden aber oft unsauber und auch redundant verwendet [78, 206]. Den Begriffen ist gemein, dass sie autonom ablaufende, strukturbildende Prozesse beschreiben.

Es existieren verschiedene Versuche der Differenzierung beider Begrifflichkeiten. So ist eine oft verwendete Abgrenzung im Bereich thermodynamischer Systeme, dass "Selbstorganisation" thermodynamische Nicht-Gleichgewichtssysteme bezeichnet, während "Selbstassemblierung" strukturbildende, dem thermodynamischen Gleichgewicht zustrebende Systeme charakterisiert [78]. Shalizi gibt in [173] eine informationstheoretisch motivierte Definition von selbstorganisierenden Systemen, als (autonome) Systeme deren informationstheoretische Komplexität mit der Zeit zunimmt. Diese Definition wird auch von Wolf in [206] zur Abgrenzung der Begriffe "Selbstorganisation" und "Emergenz" verwendet.

Die Diskussionen der Anwendung der beiden Begrifflichkeiten "Selbstorganisation" und "Selbstassemblierung" finden aber primär in der Diskussion statistisch behandelter, physikalischer oder chemischer Systeme statt, deren Verhalten analysiert und charakterisiert werden soll [19]. Parallel dazu gibt es Anwendungen im (Software-)Engineering, welche gezielt die Phänomene der Strukturbildung in autonomen Systeme zur Lösung vorgegebener Aufgaben verwenden [181, 172, 185, 126, 73]. Die Einzelsysteme werden dabei so ausgelegt, dass ihr überlagertes Verhalten eine vorgegebene Aufgabe autonom erfüllt. Nach dem subjektiven Eindruck des Autors wird in diesem Zusammenhang des Öfteren der Begriff "Selbstorganisation" ohne nähere Reflexion der eigentlichen Begriffsdefinition und Bedeutung verwendet.

Ein solches dynamisches System wird im Laufe dieser Arbeit für die autonome Bestimmung des Lösungspfades, als Alternative des in Kapitel 2.3.6 dargestellten graphenalgorithmischen Lösungspfadgenerators, vorgeschlagen (Kapitel 5.1). Dieser autonome Lösungspfadgenerator soll im weiteren Verlauf der Arbeit als *selbstassemblierender Lösungspfadgenerator* bezeichnet werden. Diese Benennung folgt der Definition der Selbstassemblierung in [203]:

"Self-assembly is the autonomous organization of components into patterns or structures without human intervention."

In [203] ist die Selbstassemblierung in verschiedenen Anwendungen und Beispielsystemen dargestellt. Die dargestellten Systeme sind aber alle materieller Natur. Die Begrifflichkeit der Selbstassemblierung soll in diesem Sinne auf die Benennung des nicht-materiellen Lösungspfadgenerators übertragen werden. Dies soll unterstreichen, dass das in dieser Arbeit vorgeschlagene und auf der mikroskopischen Ebene der den Lösungspfad bildenden Objekte definierte Regelwerk eine makroskopische Ordnungsstruktur hervorbringt. Die Dynamik des Regelwerkes endet in einem stationären Zustand ohne mikroskopische Zustandsänderungen. Dieser ähnelt, in der Analogie zu physikalischen Systemen, eher einem Gleichgewichtssystem, was in der konsistenten Charakterisierung des Regelwerkes auf den Begriff der "Selbstassemblierung" führt.

Teil II.

Wissensbasierter Entwurf

3. Entwurfssprache zur SCR-Systementwicklung

In Rahmen dieser Arbeit sollen SCR-Systeme wissensbasiert, unter Verwendung des in Kapitel 2.3 dargestellten Entwurfsparadigmas der graphenbasierten Entwurfssprachen, entworfen werden. Dazu wird in diesem Kapitel eine Entwurfssprache zur SCR-Systementwicklung dargestellt, die im Rahmen dieses Dissertationsvorhabens vom Autor entwickelt wurde und sich nun in dieser Form bereits in industrieller Anwendung im Rahmen verschiedener kommerzieller Projekte befindet.

3.1. Lösungsweg und Methodik

Wie in Kapitel 2.1.2 dargestellt ist das wichtigste Bewertungskriterium eines SCR-Systems die Stickoxid-Reduktion, da sie gerade den Einsatz eines SCR-Systems notwendig macht. Damit wird die Berechnung des Wirkungsgrades im Allgemeinen zum alles entscheidenden Bewertungsvorgang des SCR-Systementwurfes. Der zu minimierende Druckverlust über das SCR-System ist eine weitere wichtige Bewertungsgröße. Der Druckverlust hat einen negativen Einfluss auf den Wirkungsgrad des vorgeschalteten Verbrennungsmotors.

Der Entwurf von SCR-Systemen ist nach der Klassifizierung von Kapitel 2.2.2 in die Kategorie des Bottom-Up-Entwurfs einzuordnen, da der Wirkungsgrad des Systemes nicht im Voraus bestimmt werden kann. Aufgrund der nichtlinearen Effekte des Strömungsfeldes, welche über die Verteilung des Reduktionsmittels am Katalysatoreintritt entscheidend den Wirkungsgrad bestimmen, kann es keine verlässliche a priori Wirkungsgradbestimmung des Systems geben. Die Wirkungsgradberechnung muss mit Hilfe einer 3D-CFD-Berechnung erfolgen, die einen geometrisch detaillierten Entwurf voraussetzt.

Entlang dieses Grundgedanken soll in den folgenden Abschnitten eine Entwurfssprache entwickelt werden, welche eine automatisierte Generierung, Berechnung und Bewertung von SCR-Systemen erlaubt. Diese Entwurfssprache kann dann, eine ausreichend schnelle CFD-Berechnung der Systeme vorausgesetzt, um eine Rückführungsschleife entsprechend Abbildung 2.5 zur automatisierten Optimierung des Entwurfes erweitert werden.

Die Entwurfssprache zur SCR-Systementwicklung soll sich auf das Design einsträngiger Abgassysteme beschränken. Damit sind Systeme mit Strömungsvereinigungen oder Strömungsabzweigungen ausgeklammert. Es soll aber möglich sein mehrere solcher einsträngiger Abgassysteme, z.B. von mehreren Motoren, im gleichen Bauraum und unter Berücksichtigung gegenseitiger Kollisionen zu entwerfen.

3.1.1. Funktionale Zerlegung

Die Entwurfssprache für den Entwurf von SCR-Systemen soll aus mehreren modularen Entwurfssprachen¹³ bestehen, die auf dasselbe Produktmodell beziehungsweise den selben Ent-

¹³Im Folgenden auch Teilentwurfssprachen oder Teilsprachen genannt.

wurfsgraph wirken und diesen regelbasiert aufbauen. Die Teilmodule sollen anhand der funktionalen Zerlegung in Abbildung 3.1 gebildet werden.

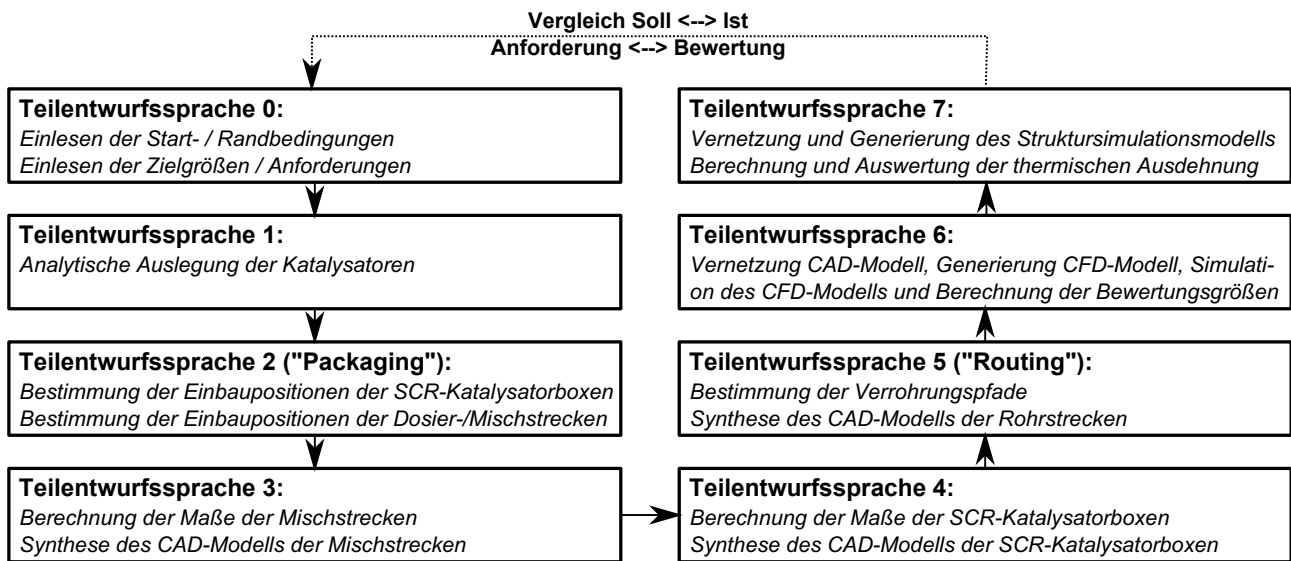


Abbildung 3.1: Funktionale Zerlegung der Entwurfssprache für den Entwurf von SCR-Systemen in modulare Teilentwurfssprachen.

Die in Abbildung 3.1 dargestellte Gesamtentwurfssprache stellt zunächst eine Entwurfssprache zur reinen Generierung konsistenter Entwürfe, auf Basis gegebener Start- und Randbedingungen, dar. Kann innerhalb der gegebenen Randbedingungen kein Entwurf erstellt werden, so soll die Entwurfssprache normal terminieren und das Ergebnis entsprechend zurückgeben.

Die gestrichelte Linie in Abbildung 3.1 deutet die Möglichkeit der Informationsrückführung der mit den Analyseprogrammen bestimmten Bewertungen an. Diese kann stellvertretend für einen einfachen Abgleich der Bewertungsgrößen mit den Anforderungsgrößen, als auch für einen Abgleich der beiden Größen mit sich anschließender regelbasierter Modifikation der Start- und Randbedingungen im Rahmen eines Optimierungsalgorithmus stehen.

Diese Informationsrückführung kann modular auf der dargestellten Generierungsentwurfssprache aufgesetzt werden. Der sinnvolle Einsatz einer automatisierten Optimierung hängt, wie sich zeigen wird, vor allem von der Simulationsdauer der CFD-Berechnung ab. Eine äußere, globale Optimierungsschleife schließt aber Schleifen innerhalb der Teilentwurfssprachen, wie beispielsweise zur Verrohrungssuche mit Hilfe der Verrohrungssprache in Kapitel 4, nicht aus.

Die folgenden Abschnitte sollen in komprimierter und vereinfachter Form die Vokabeln (UML: Klassendiagramm) erklären. Daneben wird ein vereinfachter Regelablauf der Produktionssysteme (UML: Aktivitätsdiagramm) aus den Teilentwurfssprachen von Abbildung 3.1 dargestellt.

3.1.2. Teilentwurfssprache 0: Randbedingungen und Anforderungen einlesen

Die zuerst ausgeführte Entwurfssprache ist eher technischer Natur. Sie dient zum Laden gegebener Randbedingungen und Anforderungen aus Eingabedateien in die Entwurfssprache. Die Eingabedateien lassen sich mit Excel manuell editieren oder können direkt von einem umschließenden Optimierungsalgorithmus erzeugt werden.

Abbildung 3.2 zeigt die Klassen¹⁴, welche in den Regeln des Aktivitätsdiagramm 3.3 instanziiert und mit Daten versehen werden. In der Klasse *IoPaths* werden die Laufwerksordner für das Lesen der Eingabedaten, das Schreiben der Ausgabedaten der Analyseprogramme sowie das Persistieren von Daten in den weiteren Teilentwurfssprachen gespeichert. Desweiteren können die Zugangsdaten für die Berechnungsinfrastruktur und der Pfad zur CAD-Datei des zur Verfügung stehenden Bauraumes in den Instanzen dieser Klassen hinterlegt werden.

Außerdem sind folgende Klassen für die Speicherung der für die Auslegung des SCR-Systems relevanten technischen Daten (vor allem die Klasse *Engine*) vorhanden:

- Position, Richtung und Durchmesser des Abgasauslasses des Motors
- Abgastemperatur, Abgasmassenstrom, Schadstoffemissionen (NOx)
- Indexierung der Motoren

Die Klasse *GeneralDesignParameters* dient zur Speicherung der charakteristischen Länge des vernetzten Bauraumes, des Berechnungsgitters sowie zur Hinterlegung der Anzahl der gleichzeitig zu entwerfenden Stränge. Die Positions- und Richtungsdaten der Abgasübergabestellen zur Umwelt (Auspuff-/Kaminende) werden in den Instanzen der Klasse *Environment* abgelegt. Zusätzlich enthalten diese Instanzen den Umgebungsdruck sowie die Indexierung der jeweiligen Übergabestellen, was eine Zuordnung der Abgasaustritte zu den Motoren erlaubt. Das Attribut *noxOut* in der Klasse *Environment* stellt einen Platzhalter für die in der CFD-Prozesskette berechneten Stickoxidemissionen am Abgasaustritt des Abgasreinigungssystems dar.

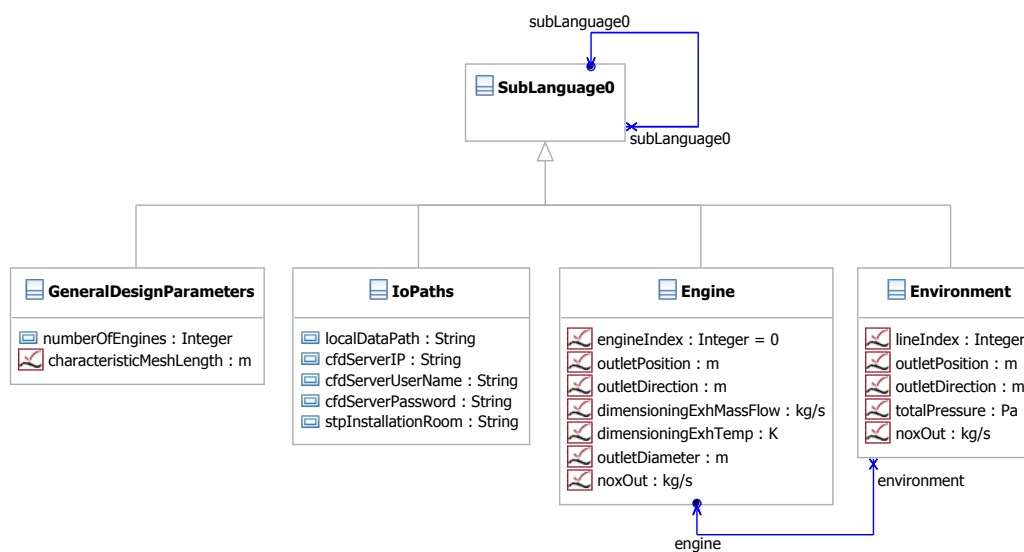


Abbildung 3.2: Klassendiagramm der Teilentwurfssprache 0 zum Einlesen der Randbedingungen und Anforderungen.

Im Aktivitätsdiagramm 3.3 werden in der ersten Regel die genannten Klassen instanziiert. In der nachfolgenden Java-Regel werden die Daten aus den entsprechenden Eingabedateien in die Attribute der zuvor instanziierten Objekte geladen.

Abbildung 3.4 zeigt beispielhaft die graphische Regel *SetEnvironment* zur Instanziiierung der *Environment*-Objekte unter gleichzeitiger Verbindung mit einer *Engine*-Instanz. Im Folgenden soll auf die explizite Darstellung einzelner Regeln aus Gründen der Übersichtlichkeit und Lesbarkeit der Arbeit verzichtet werden.

¹⁴Die jeweils äquivalenten Begrifflichkeiten "Vokabeln" = "Klassen" und "Produktionssystem" = "Aktivitätsdiagramm" sollen im Folgenden gleichberechtigt verwendet werden.

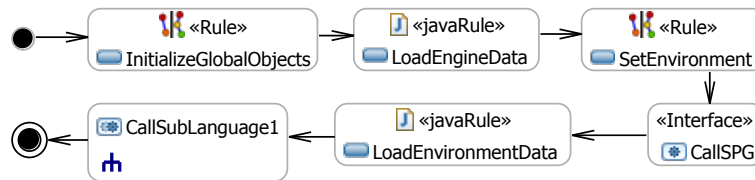


Abbildung 3.3: Aktivitätsdiagramm der Teilentwurfssprache 0 zum Einlesen der Randbedingungen und Anforderungen.

Nach der Instanziierung und Verbindung der *Environment*-Objekte mit den jeweiligen *Engine*-Objekten wird durch den Aufruf des Lösungspfadgenerators *CallSPG* die Indexierung von miteinander verbundenen *Engine*- und *Environment*-Instanzen abgeglichen. Zuletzt werden die Umgebungsdaten aus der entsprechenden Eingabedatei in die Attribute der *Environment*-Objekte mit Hilfe einer Java-Regel geladen. Abschließend wird, unter Übergabe des in der Teilentwurfssprache 0 erzeugten Entwurfsgraphen, die Teilentwurfssprache 1 aufgerufen.

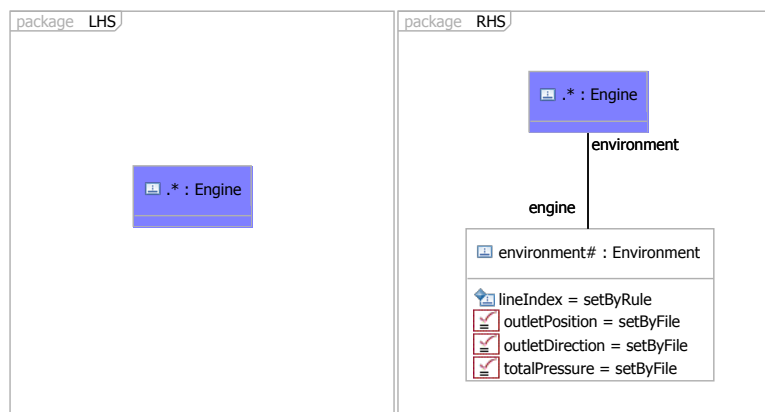


Abbildung 3.4: Exemplarische Regel des Aktivitätsdiagramm aus Abbildung 3.3.

3.1.3. Teilentwurfssprache 1: Analytische Auslegung

Die Teilsprache 1 dient der analytischen Auslegung der Katalysatorträger. Das Klassendiagramm 3.5 repräsentiert die hinterlegten Substratformen und deren jeweilige Parameter.

Im Klassendiagramm befindet sich die Klasse *Engine* der vorher ausgeführten Teilsprache 0 als sogenannter Shortcut. Diese Klasse wird im Klassendiagramm in Abbildung 3.5 über eine Assoziation der Oberklasse *SubLanguage1* und damit auch den erbbenden Klassen bekannt gemacht. Dies erlaubt einen Datenaustausch der Motordaten mit den Trägerklassen. Mit dem Verfahren der Shortcuts können prinzipiell Klassen verschiedener Klassendiagramme miteinander bekannt gemacht werden. Die auszutauschenden Daten werden in dieser Arbeit nicht explizit als UML-Interface-Elemente im Klassendiagramm definiert. Die Bekanntmachung der Klassen über direkte Assoziationen folgt den formalen Erfordernissen des Lösungspfadgenerators im Entwurfscompiler und dient zur Definition von Gleichungen aus Variablen einander bekannter Klassen.

Die Klasse *CatalystDimension* dient zur analytischen Berechnung und Speicherung der Substratgrößen. Diese erbt von der abstrakten Klasse *CatalystSubstrateResistance* die Strömungswiderstandskoeffizienten des Darcy-Forchheimer-Gesetzes [12] aus Gleichung (2.7) für die CFD-

Berechnung. Eine abstrakte Klasse kann selbst nicht instanziiert werden. Nur die Kinder einer abstrakten Klasse können instanziiert werden. Sie fungiert damit als abstrakte Schablone.

Im Gegensatz zur Klasse *CatalystDimension*, welche die berechneten Zielgrößen der Katalysatorträger enthält, sind in der Klasse *SelectedSubstrates* die konkret gewählten Daten der im Entwurf verwendeten Träger gespeichert. Diese konkret gewählte Auslegung greift auf gegebene Standardträger (Klassen *Single*Brick*¹⁵) zurück und soll möglichst nahe an den berechneten Zielgrößen der Klasse *CatalystDimension* liegen. Alle einem Abgasstrang zugeordneten Klassen beinhalten das Attribut *LineIndex* zur Zuordnung der Träger zur jeweiligen Abgaslinie.

Die verfügbaren Trägerquerschnitte seien rund, ringförmig und rechteckig (Klassen **Catalyst*). Für die jeweilige Katalysatorauslegung soll auf die nächstgelegenen Standardsubstrate (Klassen *Single*Brick*) zurückgegriffen werden. Die Klassen **Catalyst* erben von der Klasse *SelectedSubstrates* und repräsentieren so eine konkrete Katalysatorauslegung mit spezifischer Substratform, unter Verwendung von Standardsubstraten.

Bei rechteckigen Querschnitten können einzelne rechteckige Substratelemente *SingleSquareBrick* regelbasiert in mehreren Ebenen in eine rechteckige Gesamtanordnung positioniert werden, welche der Zielgröße in *Catalyst Dimension* möglichst nahe kommen soll. Das gewünschte Seitenverhältnis der rechteckigen Trägerebene kann über das entsprechende Attribut *targetWidth2Height* vorgegeben werden.

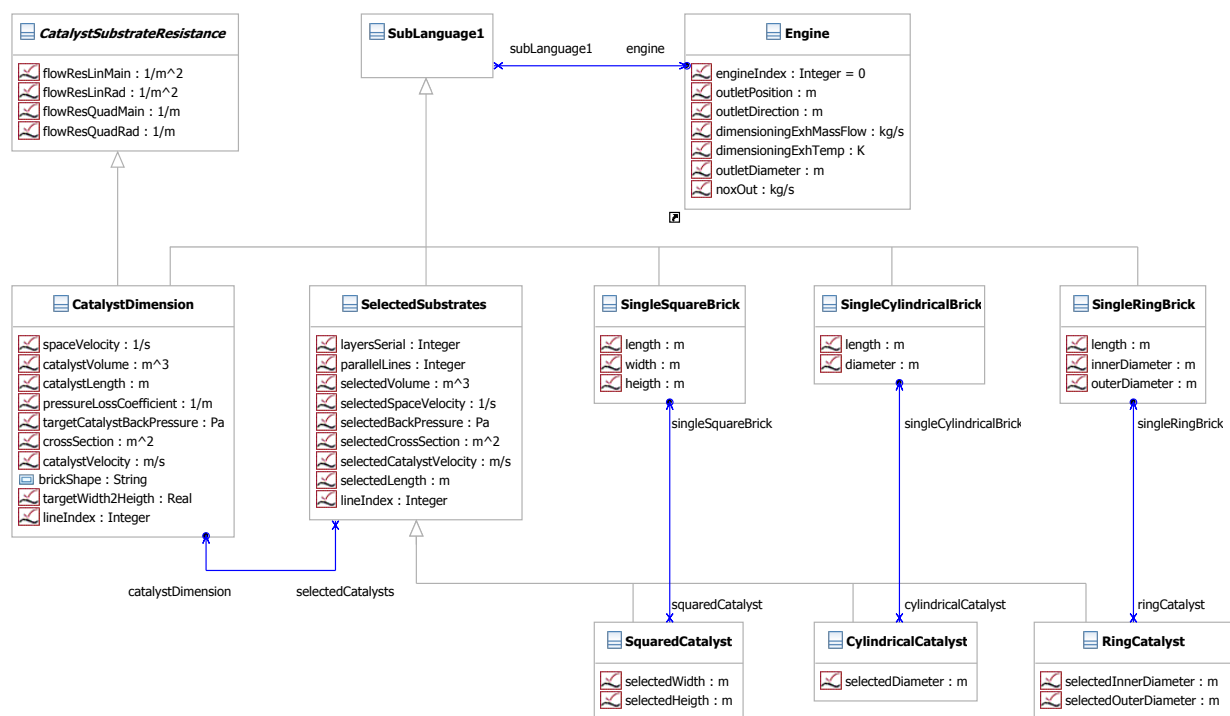


Abbildung 3.5: Klassendiagramm der Teilentwurfssprache 1 zur analytischen Auslegung der Abmaße der Katalysatorträger.

Die entsprechenden Auslegungsgleichungen zur analytischen Berechnung der Trägergrößen werden in den jeweiligen Klassen hinterlegt und können so im Lösungspfadgenerator mit angeschlossenenem Computeralgebrasystem direkt gelöst werden.

¹⁵Ein * in Beschreibungen des UML-Modells sei im folgenden als Wildcard-Symbol zu interpretieren.

Als Kennzahl für die Katalysatorgröße¹⁶ wird die Raumgeschwindigkeit als Eingangsgröße vorgegeben. In der Praxis sind die zu wählenden Raumgeschwindigkeiten auf Basis von Erfahrungswerten für verschiedene Anwendungen bekannt. Die Raumgeschwindigkeit RG berechnet sich aus dem Quotienten des durch den Katalysator strömenden Norm-Volumenstrom \dot{V}_{Kat}^o des Abgases und dem Katalysatorvolumen V_{Kat} zu (entsprechend Gleichung (2.12)):

$$RG = \frac{\dot{V}_{Kat}^o}{V_{Kat}}. \quad (3.1)$$

Auf Basis der gegebenen Motordaten, mit vorgegebenem Massenstrom \dot{m}_{Abg} , ergibt sich aus Gleichung (3.1) das Ziel-Katalysatorvolumen in Klasse *CatalystDimension* zu:

$$V_{Kat} = \frac{\dot{m}_{Abg}}{\rho^o \cdot RG} = \frac{\dot{m}_{Abg}}{1.275 \frac{kg}{m^3} \cdot RG}. \quad (3.2)$$

Mit der Druckverlustgleichung (2.8) für turbulente Strömungen lässt sich der Druckverlust Δp , bezogen auf die Katalysatorlänge L_{Kat} , mit dem längenbezogenen, dimensionslosen Druckverlustbeiwert $\zeta' = \frac{\zeta}{L_{Kat}}$ des Trägers und der Abgasgeschwindigkeit v_{Abg} unter Vernachlässigung des laminaren Termes zu

$$\frac{\Delta p}{L_{Kat}} = \frac{1}{2} \cdot \zeta' \cdot \rho_{Abg} \cdot v_{Abg}^2, \quad (3.3)$$

annähern. Aus den beiden Gleichungen (3.2) und (3.3) ergibt sich folgende Gleichung zur Bestimmung der Katalysatorlänge für einen vorgegebenen Zielgegendruck $\Delta p = \Delta p_{Ziel}$ der Träger bei einer Abgastemperatur von T_{Abg} :

$$L_{Kat} = \sqrt[3]{\frac{\Delta p_{Ziel} \cdot 2 \cdot 273K}{1.275 \frac{kg}{m^3} \cdot RG^2 \cdot T_{Abg} \cdot \zeta'}}. \quad (3.4)$$

Der erforderliche Katalysatorquerschnitt lässt sich abschließend aus den Gleichungen (3.2) und (3.4) bestimmen:

$$A_{Kat} = \frac{V_{Kat}}{L_{Kat}}. \quad (3.5)$$

Das Aktivitätsdiagramm 3.6 zeigt die Regelabfolge zur analytischen Berechnung der Abmaße der Katalysatorträger. In der ersten Regel wird für jeden Motor eine Instanz der Klasse *CatalystDimension* zur Berechnung der Zielgrößen des Trägers auf Basis der Motordaten instanziiert und mit der jeweiligen Instanz von *Engine* verbunden. In der folgenden Java-Regel werden wiederum die gegebenen Auslegungsdaten (u.a. die Ziel-Raumgeschwindigkeit, der maximale Trägergegendruck und die gewünschte Trägerform) von den Eingabedateien eingelesen. Mit dem Aufruf des Lösungspfadgenerators (*CallSPG*) werden die Zielgrößen der Träger in den Instanzen von *CatalystDimension* nach den oben dargestellten Gleichungen berechnet und die Indexierung der Abgasstränge angeglichen.

Anschließend werden den Instanzen von *CatalystDimension* die jeweiligen Instanzen der Substrate **Catalyst* sowie die der Standardsubstrate *Single*Brick* mit den entsprechenden Trägerformen, wiederum durch eine graphische Regel durch Instanziierung und Verbindung, zugeordnet. Eine programmierte Regel lädt die Daten der Standardsubstrate aus den Eingabedateien in der nachfolgenden Regel nach.

¹⁶„Katalysator“ und „Träger“ beziehungsweise „Katalysatorträger“ wird im Folgenden, wie in der Praxis üblich, äquivalent benutzt.

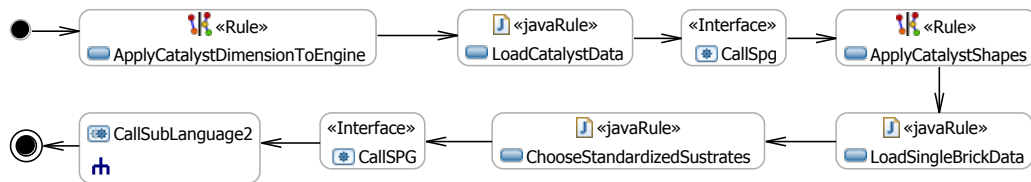


Abbildung 3.6: Aktivitätsdiagramm der Teilentwurfssprache 1 zur analytischen Auslegung der Abmaße der Katalysatorträger.

Die Java-Regel *ChooseStandardizedSubstrates* vergleicht nun die berechneten Zielgrößen der Träger in den Instanzen der Kinder von *CatalystDimension* mit den Parametern der instanziierten Standardträgern von *Single*Brick*. Für die runden und ringförmigen Substrate wird der bezüglich des Strömungsquerschnittes am nächsten gelegene Standardträger ausgewählt. Bei rechteckigen Substraten (Instanzen von *SquaredCatalyst*) wird eine rechteckige Anordnung der Standardsubstrate (Instanzen von *SingleSquareBrick*) berechnet, welche dem gewünschten Trägerquerschnitt sowie dem geforderten Seitenverhältniss des zusammengesetzten rechteckigen Trägers am nächsten kommt. Die Teilentwurfssprache 1 schließt mit dem Aufruf der nach Abbildung 3.1 nachfolgenden Entwurfssprachen und der Übergabe des Entwurfsgraphen ab.

3.1.4. Teilentwurfssprache 2: Packaging der Komponenten

In der Teilsprache 2 wird das Packaging der Komponenten des SCR-Systems berechnet. Dies umfasst die algorithmische Bestimmung der Einbaupositionen der Katalysatorboxen und der Mischstrecken, welche dann in Teilsprache 5 verrohrt werden. Das Klassendiagramm in Abbildung 3.7 gibt die dazu verwendeten Klassen wieder.

Die beiden von *Point* ererbenden Klassen, auf der rechten Seite des Klassendiagrammes, beziehen die Positionen und Richtungen des Abgasauslasses des Motors und der Abgasübergabestelle zur Umgebung über entsprechende Assoziationen. Die Klasse *LineConfiguration* enthält die für die Bestimmung der jeweiligen Komponentengrößen notwendigen Parameter. Die Ausdehnung der Träger wird über eine Assoziation zur (Ober-)Klasse *SelectedSubstrates* aus der zuvor ausgeführten Teilsprache 1 zur analytischen Auslegung der Träger bezogen. Diese Trägerabmaße werden um die geometrischen Abmessungen der Vor- und Endkammer ergänzt und ergeben so das Maß der Einhüllenden der jeweiligen Komponenten.

Für die Positionierung der Mischstrecken werden entsprechend die äußeren Maße vorgegeben. Diese Daten können entweder direkt aus den Eingabedateien bezogen oder regelbasiert auf Grund von Erfahrungswerten und physikalischen Eingabedaten bestimmt werden.

Sub2Success enthält boolesche Parameter, die den Status der Suche der Einbaupositionen repräsentieren. Die von der Oberklasse *BoxedPosition* ererbenden Klassen nehmen die gefundenen Positionen auf. Die Zieleinbaupositionen werden in Form von achsenparallelen, quaderförmigen Hüllkörpern gespeichert, welche die jeweils zu positionierenden Komponenten umschließen.

Die Klasse *PositioningData* enthält Informationen über die Reihenfolge der Einbaupositionsbestimmung der einzelnen Komponenten sowie über deren relative Lage zueinander. Beispielsweise kann eine achsenparallele Anordnung der Komponenten verschiedener Stränge mit einem vorgegebenen Abstand der Komponenten erzwungen werden. Diese Informationen können wiederum aus Eingabedateien bezogen oder regelbasiert variiert werden. Das Attribut *rel*Pos* speichert die gewünschte relative Positionierung der jeweiligen Komponente im Bauraum entlang eines Positionierungspfades zwischen Motorauslass und Auspuff-/Kaminende.

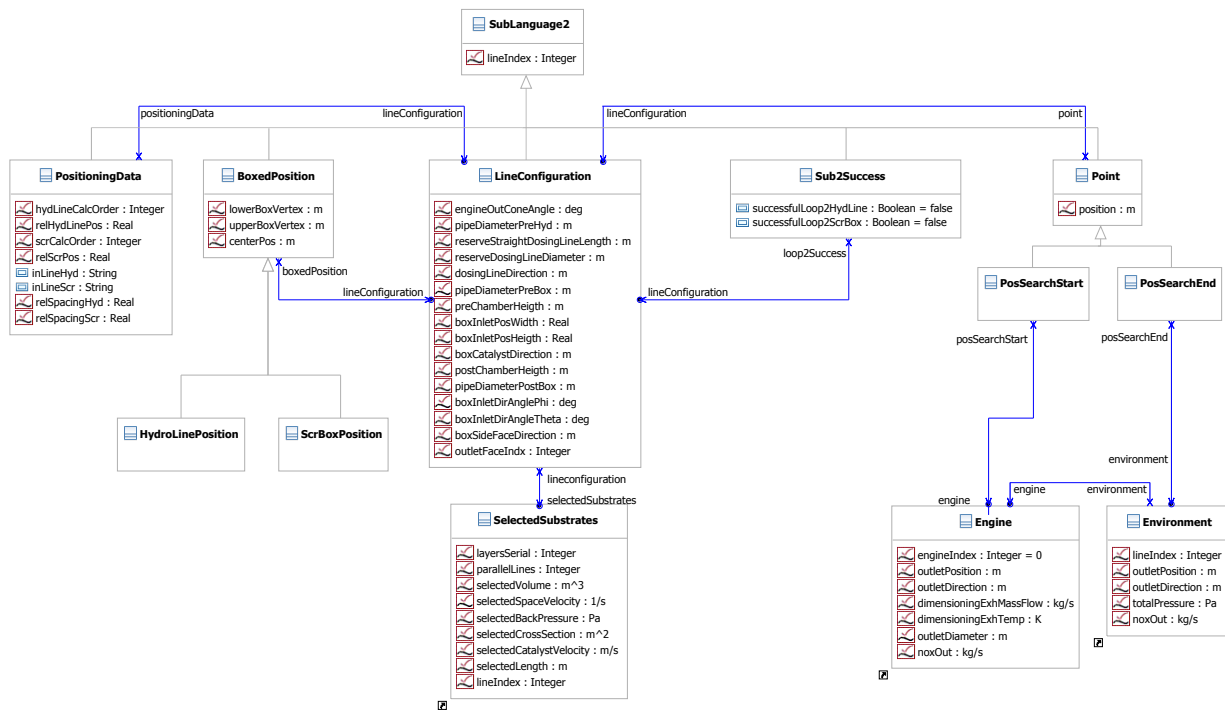


Abbildung 3.7: Klassendiagramm der Teilentwurfssprache 2 zur Positionierung der Mischstrecken und SCR-Katalysatorboxen im Bauraum (Packaging).

Der Positionierungspfad zwischen Motorauslass und Auspuff-/Kaminende wird mit Hilfe eines Pfadsuchalgorithmus auf dem diskretisierten Bauraum bestimmt. Dieser Pfad kann wie der Referenzpfad in Kapitel 4.1.4 bestimmt werden¹⁷. Dies ermöglicht, durch Verwendung einer entsprechenden Gütefunktion (siehe Gleichung (4.16)), den Pfad und damit die Einbaupositionen der Komponenten in entsprechende Regionen mit bevorzugten Eigenschaften zu bringen. In engen Bauräumen bietet sich, entsprechend Kapitel 4.1.4, ein wandferner Pfad mit maximalen Abständen zu den Bauraumgrenzen, entsprechend Gleichung (4.35), an.

Dieser Positionierungspfad wird mit der Variablen p (entspricht dem Attribut $rel*Pos$) mit $p \in [0, 1]$ parametrisiert (siehe auch Gleichung (4.20)). Somit entspricht jeder Wert von p einem Punkt $\vec{P}(p)$ auf dem Pfad. $\vec{P}(p = 0)$ entspricht der Auslassposition des Motors. $\vec{P}(p = 1)$ ist die Übergabestelle des Abgases zur Umgebung.

Die Entwurfssprache verfügt über angeschlossene CAD-Funktionalitäten, welche in OpenCascade [136] implementiert sind. Diese erlauben das Herausfiltern der Segmente eines Pfades, welche einen ausreichend großen Abstand zu den Bauraumgrenzen besitzen, um die Komponenten kollisionsfrei zu platzieren. Für die Filterung der Pfadpositionen wird auf die oben beschriebene, achsenparallele Einhüllende der jeweils zu platzierenden Komponenten zurückgegriffen.

Die Abbildung 3.8 zeigt einen solchen gefilterten Pfad in einem Bauraum mit ausgeblendeten Seitenflächen, so dass die Sicht auf das Innere des Bauraumes frei wird. Wie bereits dargestellt kann eine abhängige Ausrichtung der Komponenten verschiedener Abgasstränge (z.B. entlang einer Achse) erzwungen werden. In diesem Falle wird die Gesamt-Einhüllende der zusammengesetzten Einzelkomponenten in der Filterung des Pfades berücksichtigt.

¹⁷Zur Vermeidung von Redundanzen innerhalb der Arbeit wurden diese "technischen" Erläuterungen bewusst im formaler gestalteten Kapitel der Verrohrungssprache untergebracht.

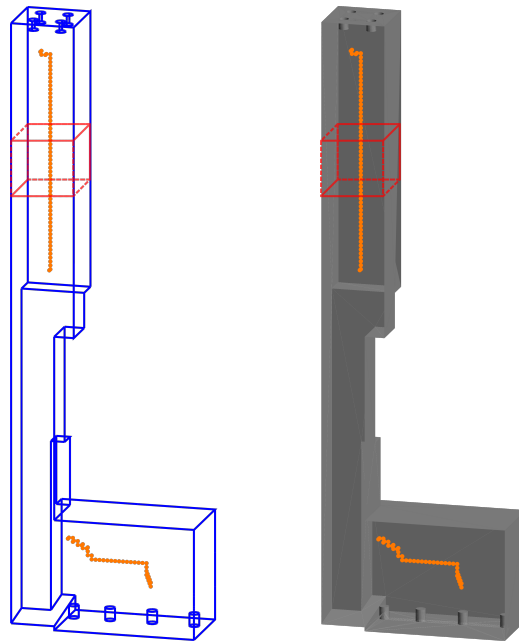


Abbildung 3.8: Gefilterte Einbaupositionen (orange), entlang eines Pfades von Motorauslass zu Auspuff-/Kaminende, mit ausreichendem Abstand zu den Bauraumgrenzen für den kollisionsfreien Einbau der Komponenten im Packaging. Eine mögliche Einhüllende der einzubauenden Komponenten ist rot dargestellt (links: Drahtmodell des Bauraumes, rechts: Flächenmodell).

Durch die Verwendung des Parameters p zur Festlegung der Einbauposition entlang eines Pfades kann der im Vorentwurf oft anzutreffende Jargon, in dem typischerweise nur von groben Umpositionierungen gesprochen wird, umgesetzt werden: So kann die Forderung, „...platziere die SCR-Katalysatorbox weiter weg vom Motor...“, über eine einfache Erhöhung des entsprechenden Parameters p der Entwurfssprache mitgeteilt werden und ein entsprechender Neuentwurf des ganzen Abgassystems getriggert werden.

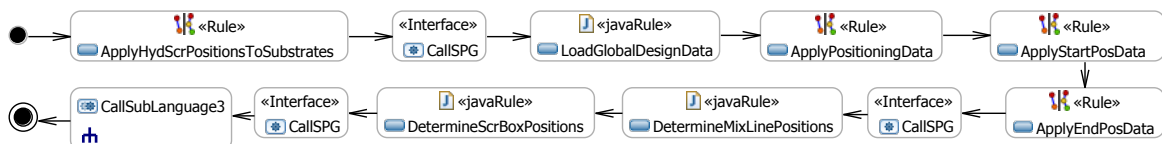


Abbildung 3.9: Aktivitätsdiagramm der Teilentwurfssprache 2 zur Positionierung der Mischstrecken und SCR-Katalysatorboxen im Bauraum (Packaging).

Das Aktivitätsdiagramm 3.9 stellt die Regelabfolge zur Umsetzung der beschriebenen Logik zur Bestimmung der Einbaupositionen dar. Die erste Regel instanziiert und verbindet die *LineConfiguration*-Objekte für jede der zu positionierenden Komponenten (Mischstrecken und SCR-Katalysatorboxen). Der anschließende Aufruf des Lösungspfadgenerators dient zum Austausch der Daten zwischen den neu verbundenen Objekten. In der folgenden Java-Regel werden die in der Eingabedatei gegebenen Daten in die betreffenden Objekte geladen (z.B. relative Positionierung, Sequenz der Positionsfindung, Abmessungen der Vor-/Endkammern, etc.).

Die folgenden drei graphischen Regeln instanziiieren die von *BoxPosition* ererbenden Klassen zur späteren Ablage der gefundenen Zielpositionen sowie die von *Point* ererbenden Klassen zur Speicherung der Positionen der Motorauslasspositionen und der Abgasübergabepositionen zur Umgebung. Letztere werden im Zuge der Instanziierung mit den bestehenden Objekten, welche

die Positionsdaten enthalten, entsprechend den Assoziationen des Klassendiagrammes verbunden und die Daten über den folgenden Aufruf des Lösungspfadgenerators ausgetauscht.

Die aus Programmcode bestehenden Regeln *Determine*Positions* führen dann die beschriebene Logik zur Bestimmung der Pfadpositionen aus: Zunächst wird der Bauraum diskretisiert und der entsprechende Pfadsuchalgorithmus, für die Positionierung der jeweiligen Komponente, in der in den Instanzen der *PositioningData*-Klasse definierten Reihenfolge ausgeführt. Die Einhüllenden der bereits positionierten Komponenten werden vor den folgenden Positionierungsdurchläufen jeweils vom Bauraum abgezogen.

Die Abmaße der Einhüllenden der Komponenten werden auf Basis der Parameter in den Instanzen der *LineConfiguration*-Klasse bestimmt und der Pfad entsprechend gefiltert. Die dem relativen Positionsparameter p (entspricht dem Attribut *rel*Pos*) nächstliegende Position des gefilterten Pfades wird bestimmt und in die Felder der Instanzen der von *BoxedPosition* erbenenden Klassen geschrieben.

Je nach Erfolg der Bestimmung der Einbaupositionen werden die entsprechenden booleschen Variablen in den Instanzen der *Sub2Success*-Klasse geschrieben. Bei nicht-erfolgreicher Suche einer Einbauposition wird die Entwurfssprache angehalten und mit entsprechender Meldung beendet. Im Falle der Verwendung eines umschließenden Optimierungsregelwerks kann dann ein neuer Entwurf mit geänderten Eingabedaten gestartet werden.

Der abschließende Aufruf des Lösungspfadgenerators dient wiederum zum Datenaustausch zwischen den Objekten. Die letzte Regel ruft die nachfolgende Teilentwurfssprache 3 auf und übergibt den aktuellen Entwurfsgraph.

3.1.5. Teilentwurfssprache 3: Geometrie der Mischstrecken

In der Teilentwurfssprache 3 soll die Geometrie und das CAD-Modell der Mischstrecke entworfen und generiert werden. Dabei wird ein Flächenmodell der Mittelflächen der Rohrwände erstellt. Eine Mischstrecke soll eine gerade Rohrstrecke darstellen, die aus folgenden Grundelementen zusammengesetzt werden kann: gerade Rohrstücke, Konen, Mischerelemente und Hydrolysekatalysator. Die äußeren Abmaße der Mischstrecke werden von den Instanzen der Klasse *LineConfiguration* der vorher ausgeführten Entwurfssprache bezogen, in denen die Einhüllenden der Mischstrecken für die Bestimmung der Einbaupositionen verwendet wurden.

Im Klassendiagramm 3.10 findet sich entsprechend ein Shortcut zur Klasse *LineConfiguration* wieder. Die darin gespeicherten Grobabmaße der Mischstrecke können über die entsprechende Assoziation an die Klasse *PositionHydLine* übergeben werden. Die beiden von *PositionHydLine* erbenenden Klassen beziehen die in Teilsprache 2 bestimmte Einbauposition von der Klasse *HydroLinePosition* der vorherigen Teilsprache. Damit können in den Klassen **PositionHydLine* mit Hilfe hinterlegter Gleichungen die Ein- und Abströmungspositionen der Mischstrecken inklusive der zugehörigen Ausrichtungen im absoluten Koordinatensystem des Einbauraumes an den Einbaupositionen bestimmt werden.

Die Klasse *HydLineElements* bezieht analog die äußeren Abmaße der Mischstrecke sowie die Rohrdurchmesser der Verrohrungsabschnitte vom Motor zur Mischstrecke und von der Mischstrecke zu den SCR-Katalysatorboxen von *LineConfiguration*. Auf Basis dieser Referenzmaße (Abmaße der Einhüllenden und die Eintritts- und Austritts-Rohrdurchmesser) wird ein relatives Maßkonzept für die Modellierung der Mischstreckengeometrie verfolgt. Die oben genannten

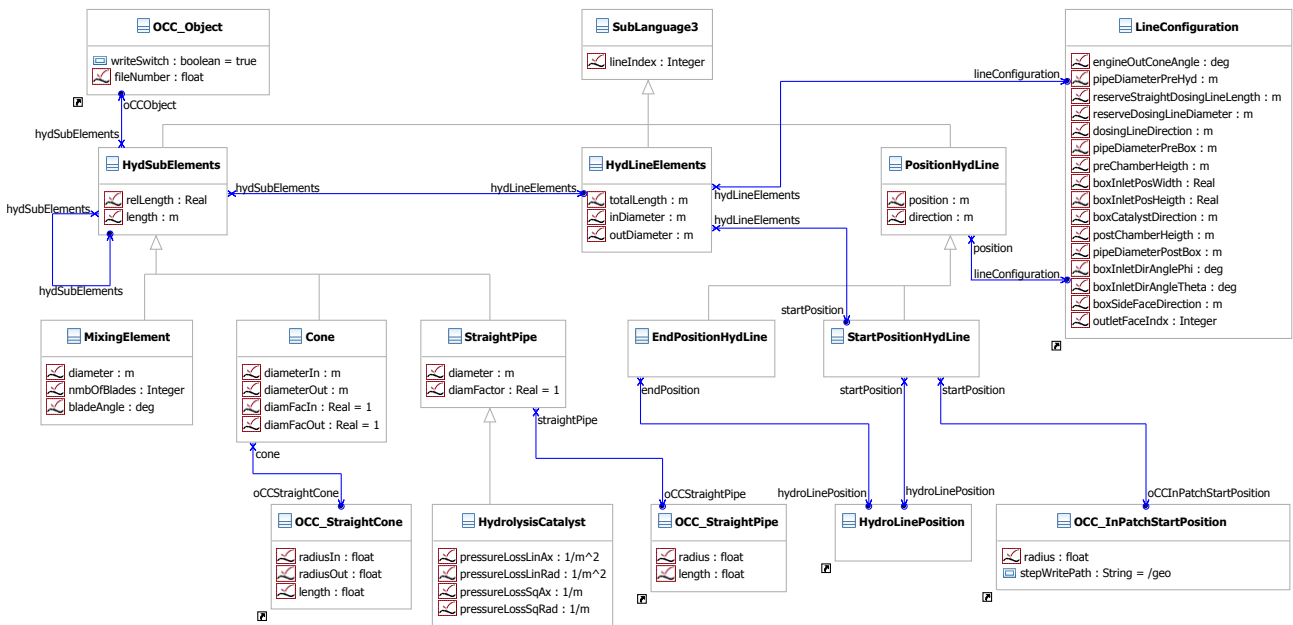


Abbildung 3.10: Klassendiagramm der Teilentwurfssprache 3 zur Generierung der Geometrie und des CAD-Modells der Mischstrecken.

geometrischen Grundelemente der Mischstrecke werden in Form relativer Faktoren, bezogen auf die genannten Referenzdimensionen, bemaßt.

Die Grundelemente der Mischstrecke erben von *HydSubElements* das Attribut *relLength*, welches die relative Länge des Elements, bezogen auf das Längenmaß der Einhüllenden (in Strömungsrichtung), beschreibt. Das geerbte Attribut *length* nimmt dann die auf Basis der angegebenen relativen Länge berechnete absolute Länge des Elements auf. Die Summe der relativen Längen der Grundelemente einer Mischstrecke muss eins ergeben. Dies kann durch eine entsprechende Regel sichergestellt werden. Die Durchmesser der Grundelemente *diam** werden analog als relative Größen dargestellt. Abbildung 3.11 zeigt drei beispielhafte Mischstrecken, aufgebaut aus den Grundelementen.

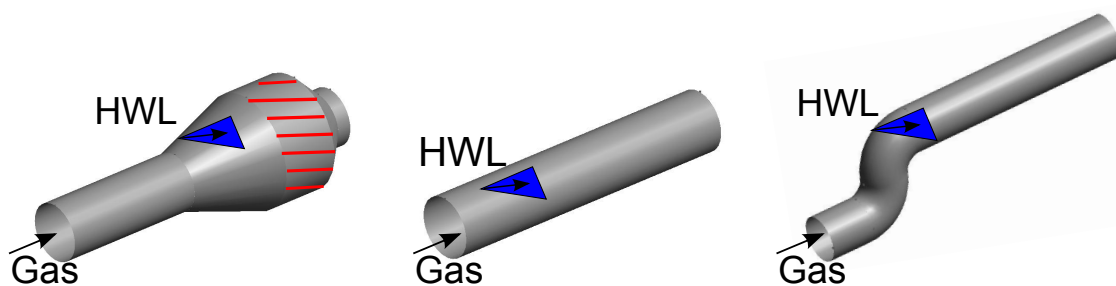


Abbildung 3.11: Beispielhafte geometrische Konfigurationen der Mischstrecken aufgebaut aus den Grundelementen. HWL-Dosierung in blau dargestellt. Von links nach rechts: Mischstrecke mit Hydrolysekatalysator (rot), gerades Rohr als Mischstrecke und Eindosierung in S-Bogen.

Die jeweiligen Grundelemente sind über Shortcuts mit den Klassen des OpenCascade-Plug-In verbunden. Das Plug-In stellt strömungsführende Grundelemente in parametrisierter Form zur Verfügung und kann aus miteinander verbundenen Elementen ein CAD-Modell der zusammengesetzten Bauteile erzeugen. Die Klasse *OCC_InPatchStartPosition* nimmt die absolute

Platzierung des ersten Geometrieelementes auf. Die weiteren Elemente werden relativ zu diesem Element platziert.

Miteinander verbundene und die Grundelemente repräsentierende Instanzen der Klassen *OCC_** des OpenCascade-Plug-In werden im Plug-In automatisch geometrisch verbunden: Am Auslass des Vorgängerelements wird das nachfolgende Element ausgerichtet und positioniert. Mit dem Lösungspfadgenerator wird die Konsistenz der Durchmesser miteinander verbundener Elemente sichergestellt und so eine geschlossene Geometrie gewährleistet.

In [168] wurde eine universelle Geometrieentwurfssprache entwickelt, die einen generischeren Ansatz der Geometrieerstellung verfolgt. Dieser Ansatz stand aber erst zum Abschluss der vorliegenden Arbeit zur Verfügung und konnte so nicht eingesetzt werden, soll aber dem interessierten Leser als weiterführende Literatur zum Thema der Geometriegenerierung mit Entwurfssprachen nahegelegt werden.

Abbildung 3.12 stellt das Aktivitätsdiagramm der vorliegenden Teilentwurfssprache dar. Die erste Regel instanziiert die Klassen **PositionHydLine* zur Speicherung der Einlass- und Auslasspositionen und Richtungen der Mischstrecke. Sie verbindet diese mit den Instanzen der Klasse *HydroLinePosition*, welche die Positionierung der Komponenten im Bauraum enthält. Ebenso mit den Instanzen von *LineConfiguration*, welche die Abmaße der Einhüllenden der Mischstrecke beherbergt. Im folgenden Aufruf des Lösungspfadgenerators werden die Daten ausgetauscht und in der anschließend aufgerufenen Java-Regel die Einlass- und Auslasspositionen und Richtungen der Mischstrecken berechnet.

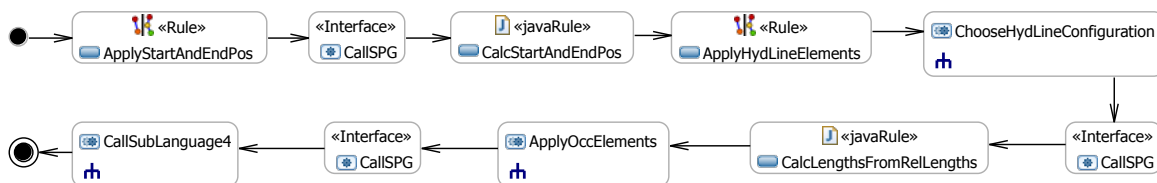


Abbildung 3.12: Aktivitätsdiagramm der Teilentwurfssprache 3 zur Generierung der Geometrie und des CAD-Modells der Mischstrecken.

In der darauf folgenden graphischen Regel *ApplyHydLineElements* wird an jede Mischstrecke (Instanz von *LineConfiguration*) eine Instanz der Klasse *HydLineElements* zur Speicherung der Referenzmaße angehängt. Die nachfolgende Aktivität *ChooseHydLineConfiguration* stellt eine Sub-Aktivität dar, die wiederum mehrere Regeln für die Instanziierung verschiedener Mischstreckenkonfigurationen (gerades Rohr, Konus-Hydrolysekatalysator-Konus, gerades Rohr mit Mischer) enthält.

Abbildung 3.13 stellt das entsprechende Aktivitätsdiagramm dieser Sub-Aktivität dar. Im Entscheidungsknoten (Raute) werden die Einbauregeln entsprechend der in den Eingabedateien vermerkten Mischstreckenkonfigurationen aufgerufen.

Der nachfolgende Aufruf des Lösungspfadgenerators dient zum Austausch der Daten. In der anschließenden Java-Regel werden die oben dargestellten absoluten Maße der Rohrelemente der Mischstrecke auf Basis der in der jeweiligen Mischstreckenkonfiguration hinterlegten relativen Maßangaben berechnet. Die Sub-Aktivität *ApplyOccElements* enthält Regeln zur Instanziierung der OpenCascade-Klassen, welche die Geometrieparameter zur Ableitung des CAD-Modells erhalten. Die entsprechenden Daten werden mit dem abschließenden Aufruf des Lösungspfadgenerators ausgetauscht. Das vorliegende Aktivitätsdiagramm endet mit dem Aufruf der nachfolgenden Teilentwurfssprache 4 und der Übergabe des Entwurfsgraphen.

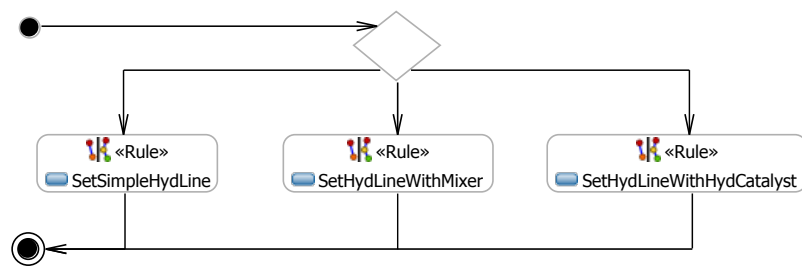


Abbildung 3.13: Subaktivität zur Auswahl einer Mischstreckenkonfiguration mit Hilfe eines Entscheidungsknotens (Raute).

3.1.6. Teilentwurfssprache 4: Geometrie der Katalysatorboxen

In Teilentwurfssprache 4 soll die Geometrie und das CAD-Modell der SCR-Katalysatorboxen generiert werden. Es wird wie bei der Geometrieerstellung der Mischstrecken ein Flächenmodell der Wände erstellt. Wie im Abschnitt zur analytischen Katalysatorauslegung beschrieben, sollen runde, ringförmige und rechteckige Träger in der SCR-Katalysatorbox untergebracht werden können (Abbildung 3.14). Die äußeren Abmaße der Katalysatorbox werden wieder von den Instanzen der Klasse *LineConfiguration* der Packaging-Entwurfssprache bezogen. In ihnen werden die Einhüllende der Katalysatorbox aus den berechneten Trägergrößen von Teilsprache 1, inklusive einer Längenzugabe für die Vor- und End-Kammern der Katalysatorbox, bestimmt.

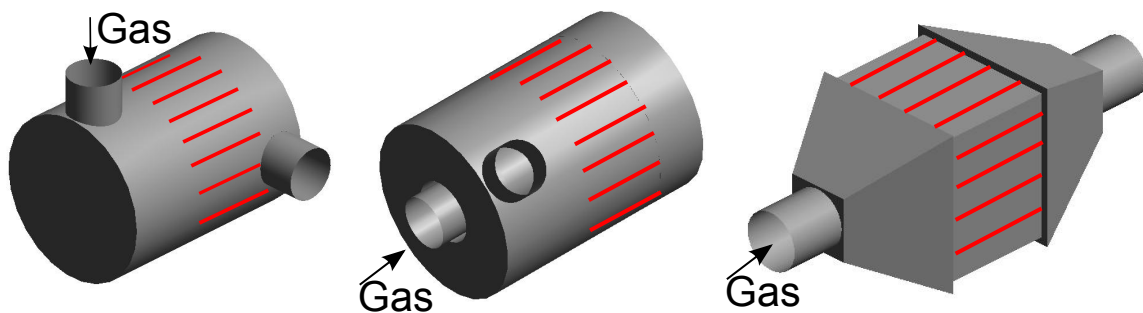


Abbildung 3.14: Geometrische Varianten der SCR-Katalysatorboxen. Katalysatorträger sind in rot angedeutet. Von links nach rechts: Zylindrischer Katalysator, ringförmiger Katalysator und rechteckiger Katalysator. Geometrische Parametrisierung der Katalysatorboxen können aus dem Klassendiagramm in Abbildung 3.15 entnommen werden.

Das Klassendiagramm 3.15 spiegelt im Wesentlichen dieselbe Logik wie in der vorhergehenden Teilsprache wieder. Die Abmaße der Katalysatorboxen sowie die in Teilsprache 2 berechnete Positionierung der Boxen soll in den von *PositionScrBox* ererbenden Klassen über Assoziationen auf die Shortcuts der Klassen *LineConfiguration* beziehungsweise *ScrBoxPosition* bezogen werden. Die Instanzen der Kinder von *PositionScrBox* sollen analog die Ein- und Abströmungspositionen und Richtungen der Katalysatorboxen speichern.

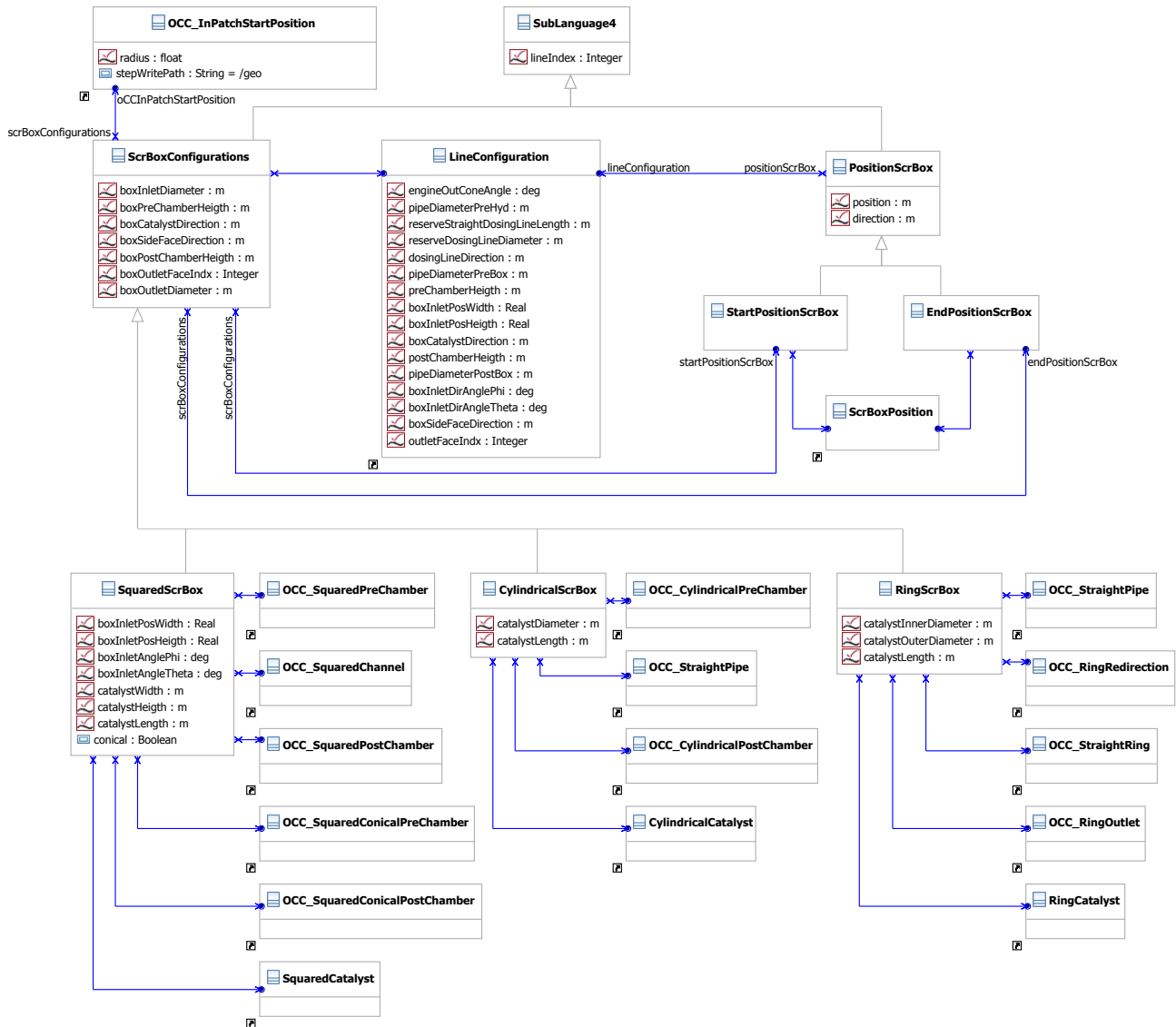


Abbildung 3.15: Klassendiagramm der Teilentwurfssprache 4 zur Generierung der Geometrie und des CAD-Modells der Katalysatorboxen. Die Attribute der Shortcuts zu den OpenCascade-Klassen sind verdeckt. Ausgewählte Bezeichner der Assoziationen sind aus Gründen der Darstellbarkeit ausgeblendet.

Die Klasse *ScrBoxConfigurations* enthält Attribute für eine von der Trägerform unabhängigen Beschreibung der gewünschten Konfiguration der Katalysatorboxen. Darin werden die Maße der Vor- und Endkammer, die Ausrichtung der Träger im Raum, sowie die Richtung des Abgasauslasses der Endkammer bestimmt.

Die auf die Trägerformen spezialisierten Kinder **ScrBox* dieser Klasse enthalten ergänzende Geometrie-Attribute der jeweiligen Trägerformen. Die Klassen **ScrBox* besitzen Assoziationen auf die zum Aufbau der entsprechend geformten Katalysatorboxen benötigten Klassen der geometrischen Grundelemente *OCC_** des OpenCascade-Plug-In. Entlang der Assoziationen werden wiederum die entsprechenden Parameterwerte ausgetauscht.

Entsprechend weist auch das Aktivitätsdiagramm 3.16 die ähnlichen Schritte wie die vorher ausgeführte Teilsprache zur Geometriegenerierung der Mischstrecken auf. In der ersten Regel werden die Klassen **PositionScrBox* zur Speicherung der Einlass- und Auslasspositionen und Richtungen instanziiert. Diese Objekte werden gleichzeitig mit den Instanzen der Klasse *HydroLinePosition*, sowie den Instanzen von *LineConfiguration*, zum Austausch der Maße der Einhüllenden und der in Teilsprache 2 bestimmten Einbauposition, verbunden.

Die folgenden drei Subaktivitäten enthalten Regeln zur Generierung und Verbindung der Instanzen, der von *ScrBoxConfigurations* erbinden, trägerformspezifischen Klassen, mit den Objekten der Klassen *LineConfiguration* und **PositionScrBox*. Die nachfolgende Java-Regel *SetOccFileNumbers* dient zur Nummerierung der einzelnen geometrischen Objekte für den Aufruf des OpenCascade-Plug-Ins.

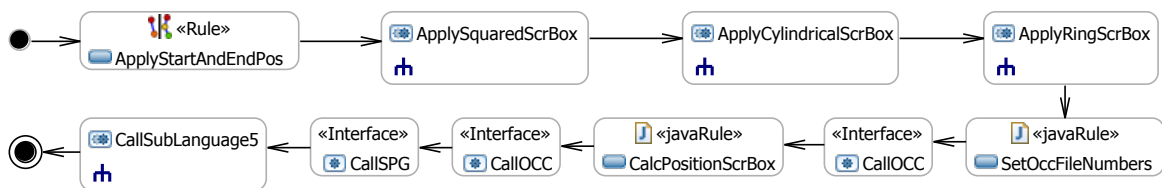


Abbildung 3.16: Aktivitätsdiagramm der Teilentwurfssprache 4 zur Generierung der Geometrie und des CAD-Modells der Katalysatorboxen.

Die Oberflächensegmente des Mittelflächenmodells müssen einzeln, für das spätere Aufbringen der Randbedingungen in der CFD-Berechnung, zugänglich sein und werden deshalb in eigenen Geometriedateien gespeichert. Für eine prinzipielle Darstellung des Zusammenspiels von Geometrierepräsentation beziehungsweise Erzeugung und dem Aufbringen von Randbedingungen in nachgeschalteten Prozessketten sei an dieser Stelle auf die Arbeit von Dungs [54] verwiesen.

Im Gegensatz zur geraden, entlang einer Achse verlaufenden Mischstrecke können diese Positionen und Richtungen aber nicht im Voraus bestimmt werden. Aus diesem Grund wird das OpenCascade-Plug-In *CallOCC* im Aktivitätsdiagramm 3.16 zweimal aufgerufen: Der erste Aufruf dient zur Generierung der Katalysatorbox am Koordinatenursprung mit den gewünschten Maßen zur Bestimmung der relativen Position von Ein- und Ausströmungsöffnung sowie zur Bestimmung der genauen Gesamtabmaße der Katalysatorbox. Auf Basis dieser Information wird in der Java-Regel *CalcPositionScrBox* die generierte Box in die im Packaging (Teilsprache 2) bestimmte Einhüllende eingepasst und die entsprechende Translation der Box vorgenommen.

Unter Berücksichtigung dieser Translation wird die Katalysatorbox nun durch den zweiten Aufruf des OpenCascade-Plug-In *CallOCC* an der Einbauposition neu generiert. Der Aufruf des Lösungspfadgenerators dient zum Austausch der Daten und im Speziellen zur Speicherung der Ein- und Ausströmungsposition und Richtung in die Instanzen der Klassen **PositionScr-*

Box anhand der Rückgabedaten des OpenCascade-Plug-In. Die letzte Regel ruft wiederum die nachfolgende Teilentwurfssprache 5 auf und übergibt den aktuellen Entwurfsgraph.

3.1.7. Teilentwurfssprache 5: Routing der Rohrstrecken

Die Teilentwurfssprache 5 stellt im Wesentlichen eine generische Schnittstellensprache dar. Sie dient zur Extraktion der für das Routing beziehungsweise für die Generierung des CAD-Modells benötigten Daten aus dem Entwurfsgraph. Für die Bestimmung der Verrohrungen der folgenden drei zu betrachtenden Rohrabschnitte:

- Motor bis Mischstrecke
- Mischstrecke bis Katalysatorbox
- Katalysatorbox bis Auspuff-/Kaminende

müssen mindestens die jeweiligen Start- und Endpositionen und Richtungen sowie der Rohrdurchmesser bekannt sein. Die Verrohrung soll dann in einer spezialisierten, modularen Entwurfssprache auf Basis dieser Eingabebedingungen erstellt und als CAD-Datei im STEP-Format zurückgegeben werden. Die Klasse *GenericPipeWork* des Klassendiagrammes in Abbildung 3.17 nimmt die entsprechenden Eingabedaten, ergänzt um das Attribut *name* zur eindeutigen Bezeichnung der entsprechenden Rohrstrecke, auf.

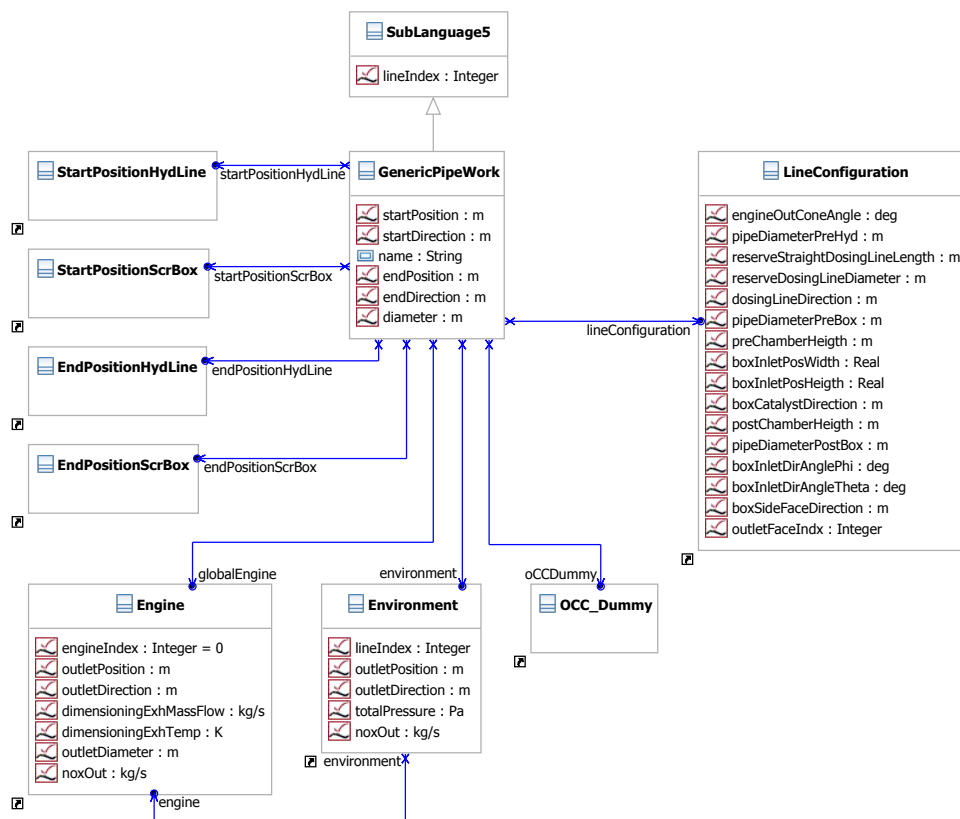


Abbildung 3.17: Klassendiagramm der Teilentwurfssprache 5 zum Auf- und Rückruf der Verrohrungssprache von Kapitel 4. Ausgewählte Bezeichner der Assoziationen sind aus Gründen der Darstellbarkeit ausgeblendet.

Die weiteren Klassen des Klassendiagrammes stellen wiederum Shortcuts auf die Klassen vorher ausgeführter Teilsprachen dar, um über Assoziationen die oben genannten Eingabedaten des

Routings aus dem Entwurfsgraph zu beziehen. Die Klassen *StartPosition** und *EndPosition** stellen die Koordinaten und Richtungen der Ein- und Auslässe der platzierten Komponenten zur Verfügung. Entsprechend werden die Positionen und Richtungen des Motorauslasses und der Abgasübergabe zur Umgebung von den Klassen *Engine* und *Environment* bezogen.

Die jeweiligen Durchmesser der Rohrstrecken werden von der Klasse *LineConfiguration* zur Verfügung gestellt. Der Shortcut auf die Klasse *OCC_Dummy* dient zur Bekanntmachung der von der spezialisierten Verrohrungssprache erzeugten CAD-Datei gegenüber dem OpenCascade-Plug-in. Der *OCC_Dummy* stellt damit ein Platzhalter für eine außerhalb des OpenCascade-Plug-in erzeugten CAD-Datei dar, welche aber bei der Generierung der gesamten CAD-Baugruppe im OpenCascade-Plug-in mit eingefügt wird.

Das Aktivitätsdiagramm 3.18 stellt die Regelabfolge zur Synthese der Rohrstrecken dar. Die ersten drei Regeln instanziiieren die Klasse *GenericPipeWork* und verbinden die erzeugte Instanz mit den Objekten der vorher ausgeführten Teilsprachen, welche die für den jeweiligen Rohrabschnitt benötigten Eingabedaten enthalten. Die Daten zwischen den Instanzen werden ebenfalls innerhalb der Java-Regeln ausgetauscht.

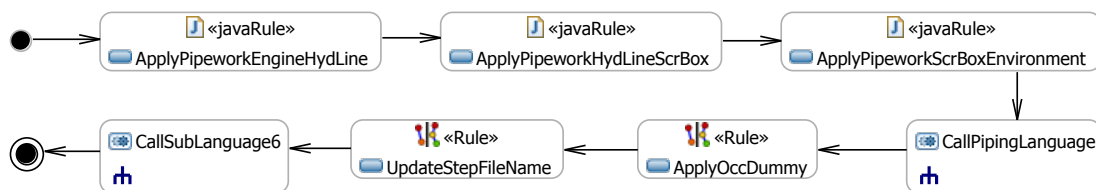


Abbildung 3.18: Aktivitätsdiagramm der Teilentwurfssprache 5 zum Auf- und Rückruf der Verrohrungssprache von Kapitel 4.

Die folgende Subaktivität *CallPipingLanguage* enthält den eigentlichen Aufruf der spezialisierten Verrohrungssprache, welche die Rohrstrecke im Bauraum routet und das CAD-Modell der Verrohrung erstellt. Diese indirekte Aufrufmethode über die zwischengeschaltete Teilsprache 5, die im Grunde eine standardisierte Schnittstelle darstellt, wird auf Grund der verschiedenen zur Verfügung stehenden Verrohrungsprinzipien gewählt.

Diese Architektur soll ein einfaches Umschalten zwischen Verrohrungsalgorithmen sowie eine manuelle Verrohrung auf Basis der in den Instanzen von *GenericPipeWork* gespeicherten Randbedingungen erlauben. Dazu kann in der Subaktivität *CallPipingLanguage* die Ausführung der Entwurfssprache pausiert und die entsprechenden Randbedingungen in eine Tabelle für die Verwendung im manuellen Routing geschrieben werden. Nach erfolgter manueller Konstruktion der Verrohrung wird die Ausführung der Entwurfssprache wieder aufgenommen.

Im Vorfeld der Arbeit wurden unter anderem Verrohrungssprachen für die Synthese rechtwinkliger Rohrstrecken, basierend auf den in Kapitel 2.4.1 dargestellten Pfadsuchalgorithmen, entwickelt. Diese Ansätze sind aber nicht in der Lage nicht-rechtwinklige Rohrstrecken, wie sie in engen Fahrzeugbauräumen benötigt werden, zu synthetisieren. Auf Grund von Laufzeitvorteilen könnten solche rechtwinkligen Verrohrungsalgorithmen aber beispielsweise in der Verrohrung von verfahrenstechnischen Anlagen vorteilhaft eingesetzt werden.

In der vorliegenden Entwurfssprache soll die im Rahmen dieser Arbeit entwickelte und in Kapitel 4 im Detail dargestellte Verrohrungssprache aufgerufen und verwendet werden. Diese benötigt für die Ausführung noch weitere Informationen, wie zum Beispiel eine Liste der zu verwendenden Rohrbögen, welche sich in der Datenstruktur dieser Verrohrungssprache im

Klassendiagramm (Abbildung 4.1) in Kapitel 4.1.1 wiederfinden und dort entsprechend aus Eingabedateien bezogen werden.

Nach der Generierung der Verrohrung wird in Regel *ApplyOccDummy* der Platzhalter für die Integration der generierten Geometrie in die OpenCascade-Baugruppe der jeweiligen Rohrstrecke initialisiert und verbunden. Anschließend werden die Dateinamen der erzeugten Geometrien angepasst. Die Teilsprache schließt mit dem Aufruf der nachfolgenden Teilentwurfssprache 6 und der Übergabe des Entwurfsgraphen ab.

3.1.8. Teilentwurfssprache 6: CFD Berechnung und Bewertung

In der Teilentwurfssprache 6 soll die zuvor generierte Geometrie vernetzt und anschließend eine CFD-Berechnung mit der Open-Source-CFD-Software OpenFOAM [137] durchgeführt werden. Die wasserdichte Geometrie des SCR-Systems liegt, wie in den Kapiteln 3.1.5 und 3.1.6 beschrieben, als eindeutig nummerierte STEP-Dateien einzelner Mittelflächensegmente der Wandungen des Abgassystems vor.

Die Teilentwurfssprache zur CFD-Berechnung soll in Form einer Prinzipdarstellung gezeigt werden, das real umgesetzte Plug-in besteht aus einer Vielzahl weiterer, detaillierterer Klassen, welche eine Einstellung aller CFD-Solver- und Vernetzungsparameter der verwendeten OpenFOAM Applikationen erlauben. Das stark reduzierte Klassendiagramm in Abbildung 3.19 spiegelt die grundsätzliche Architektur des Plug-in wieder. Dabei wird vor allem auf die Darstellung der Informationsübergabe vom CAD-Modell über die Vernetzung hin zum CFD-Modell Wert gelegt.

Die real umgesetzte Schnittstelle beziehungsweise das reale Klassendiagramm des OpenFOAM-Plug-in besitzt mehrere hundert Attribute in über hundert Klassen. Dieser Umfang wird notwendig, wenn alle in der CFD-Simulation mit OpenFOAM verwendbaren Parameter auch in der Entwurfssprache zugänglich sein sollen. Diese Implementierung einer sehr "breiten" Schnittstelle wurde gewählt, um das OpenFOAM-Plug-in möglichst generisch und damit auch in anderen Anwendungen, mit anderen Simulationsparametern, einsetzbar zu machen. Im vollständigen Klassendiagramm sind jedoch viele Attribute mit einem Standardwert, angepasst an die Berechnung von Abgassystemen, versehen, so dass nicht alle Parameter beim Ausführen der Teilsprache jeweils explizit als Eingabedaten angegeben werden müssen.

Die Vernetzung des Modells soll mit dem voll-automatisierten Vernetzungstool *snappyHexMesh*, welches Teil der OpenFOAM-Distribution ist, ausgeführt werden. Dazu müssen die Oberflächensegmente des Modells als triangulierte Oberflächen im Surface Tessellation Language (STL)-Format vorliegen, siehe Abbildung 3.20. Diese STL-Dateien werden mit Hilfe des Vernetzungstools GMSH [68], welches über die in Abbildung 3.19 von *GMSH* erbenenden Klassen mit den entsprechenden Daten versorgt wird, erzeugt. Die Instanzen der Klasse *StpSurfaceTriangulation* werden mit der jeweiligen, die STEP-Datei eines Oberflächensegmentes repräsentierenden Instanz eines Kindes von *OCC_Object* verbunden und so die STEP-Datei als Eingangsgeometrie dem Vernetzungstool GMSH bekannt gemacht.

Neben der Klasse *StpSurfaceTriangulation* stellt GMSH zwei weitere Funktionen in Form entsprechender Klassen zur Verfügung: Die Klasse *BoundingBox* repräsentiert eine Funktion zur Bestimmung der achsenparallelen Einhüllenden der triangulierten Geometrieobjekte. Mit Hilfe der Klasse *GroupSurfaces* können mehrere, mit der Instanz der Klasse verbundene, triangulierte Objekte zu einer Gruppe zusammengefasst werden. Diese Funktion wird verwendet, um die Sub-Regionen der Katalysatorträger im Berechnungsnetz zu definieren: Dazu werden die ein-

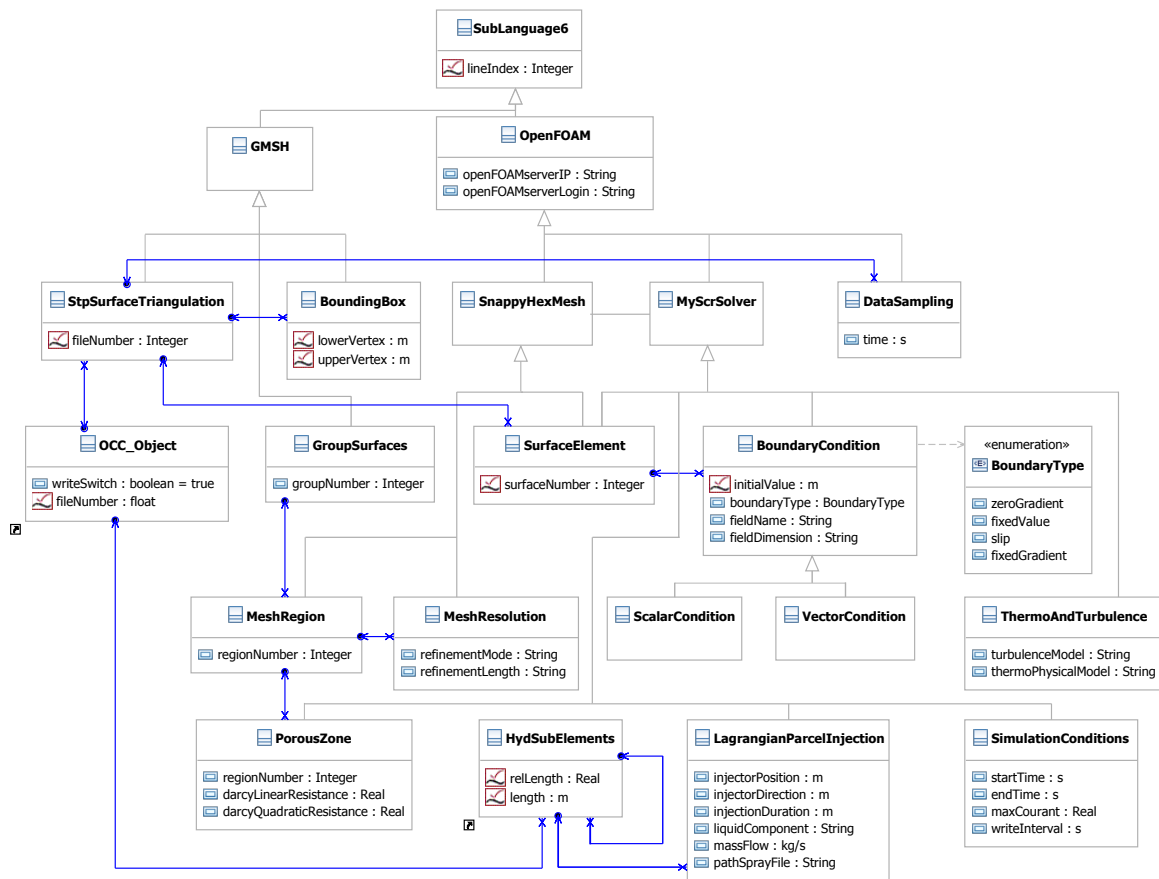


Abbildung 3.19: Vereinfachtes Klassendiagramm der Teilentwurfssprache 6 zur Vernetzung und CFD-Berechnung des SCR-Systems. Bezeichner der Assoziationen sind aus Gründen der Darstellbarkeit ausgeblendet.

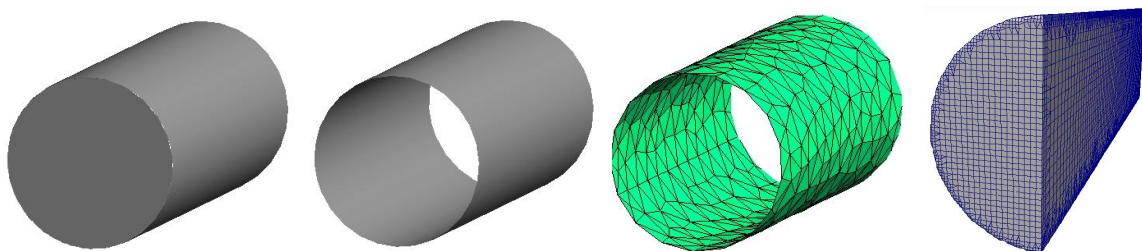


Abbildung 3.20: Geometrische Repräsentation des Vernetzungsworkflows zur Erzeugung des CFD-Modelles. Von links nach rechts: Volumenmodell, Oberflächenmodell, mit GMSH vorvernetztes Oberflächenmodell (STL-Format) zur Übergabe an den OpenFOAM Vernetzter *snappyHexMesh* und vernetztes CFD-Modell.

zelenen Oberflächensegmente des Katalysators zu einer geschlossenen Oberfläche verschmolzen. Mit Hilfe dieser Oberflächen kann dann mit dem OpenFOAM Tool *topoSet* der Bereich des Katalysatorträgers als eigene Regionen definiert werden, um später die Strömungswiderstandskoeffizienten der Träger aufzubringen.

Die von der Klasse *SnappyHexMesh* ererbenden Klassen dienen der Informationsübergabe an das Vernetzungstool *SnappyHexMesh*. Die Übergabe der triangulierten Oberflächenelemente wird über eine entsprechende Assoziation von der Klasse *SurfaceElement* zur Triangulationsklasse *StpSurfaceTriangulation* realisiert.

Daneben können Instanzen der Klasse *MeshRegion* im Zusammenspiel mit der assoziierten Klasse *GroupSurfaces* für die Definition von Sub-Regionen des Berechnungsnetzes im Vernetzungstool verwendet werden. Über die mit *MeshRegion* assoziierte Klasse *MeshResolution* kann die Auflösung des Netzes in der assoziierten Region gesteuert werden. Damit können beispielsweise auch die Ein- und Austrittsbereiche der Träger sowie die Region um die Harnstoffdosierung als Sub-Regionen definiert und regelbasiert verfeinert werden.

Die von *MyScrSolver*¹⁸ ererbenden Klassen beherbergen die Daten zur Ausführung der CFD-Simulation. Über die Klasse *SurfaceElement* werden die Oberflächensegmente sowie die entsprechenden Oberflächenbereiche des vernetzten Modelles zugänglich. Die Instanzen der Kinder von *BoundaryCondition* dienen zum Aufbringen einer Randbedingung auf die einen Oberflächenbereich repräsentierende Instanz von *SurfaceElement*.

Die Objekte der von *BoundaryCondition* ererbenden Klassen beherbergen jeweils die Randbedingung für einen Oberflächenbereich und eines der jeweils zu berechnenden physikalischen Felder (Geschwindigkeit, Druck, chemische Komponenten, Temperatur, turbulente Energie, turbulente kinetische Energie, turbulente isotrope Dissipationsrate,...). Der mathematische Typ der Randbedingung kann aus der verbundenen Liste *BoundaryType* gewählt werden (Gradienten- und Festwert-Randbedingungen). Die entsprechenden Zahlenwerte der Randbedingungen werden aus dem bestehenden Entwurfsgraph bezogen, diese Assoziationen sind aber nicht explizit im vereinfachten Klassendiagramm 3.19 dargestellt. So werden beispielsweise die Werte des Massenstroms und der Abgastemperatur von den bereits in Teilentwurfssprache 0 initialisierten Instanzen der Klasse *Engine* bezogen.

Die Klasse *LagrangianParcelInjection* beherbergt die Daten und Informationen zur Definition der Eindosierung des Harnstoffes inklusive eines Dateiverweises zum Einlesen der Spraydaten (Tropfengrößenspektrum, Tropfengeschwindigkeit) des verwendeten Injektors. Die so eingelesenen Spraydaten definieren zusammen mit den von Klasse *HysSubElements* bezogenen Positionsdaten und dem HWL-Massenstrom *massFlow* aus Klasse *LagrangianParcelInjection* die Eindosierung der HWL-Lagrange-Phase.

Die Klasse *SimulationCoefficients* enthält Informationen bezüglich der zu berechnenden Strömungsdauer sowie die, für die adaptive Bestimmung des Simulationszeitschrittes notwendige, maximale Courant-Nummer [44]. Die Instanzen von *ThermoAndTurbulence* werden zur Selektion des zu verwendenden Turbulenzmodelles sowie zur Auswahl des thermophysikalischen Modelles verwendet.

Die Klasse *DataSampling* dient der Ergebnisauswertung. Die Instanzen der Klasse können mit triangulierten Oberflächenelementen, welche Schnittebenen im Berechnungsgitter definieren, verbunden werden. Die Werte der in der Simulation berechneten Strömungsfelder werden dann

¹⁸MyScrSolver ist der Bezeichner des im Rahmen der vorliegenden Entwurfssprache für die Berechnung von AGN-Systemen implementierten Solvers, unter Verwendung der OpenFOAM-Bibliotheken .

an den so definierten Querschnitten nach der Simulation ausgelesen. So können beispielsweise die Strömungs- und Ammoniakverteilungen am Eintritt des SCR-Katalysators, über die triangulierte Stirnseiten der Träger, bestimmt werden.

Das Aktivitätsdiagramm 3.21 beschreibt schematisch die Regelabfolge zur Vernetzung, CFD-Berechnung und zum Postprocessing des SCR-Systems. Im ersten Schritt werden die Instanzen zur Triangulation der als STEP-Dateien vorliegenden Oberflächensegmente instanziiert und verbunden. Die neu instanziierten Objekte zur Vernetzung der Oberflächensegmente werden in der nachfolgenden Regel mit der Instanz der Klasse *BoundingBox* zur Bestimmung der Einhüllenden des Modelles verbunden. Die Einhüllende wird vom Vernetzungstool als Eingabegröße gebraucht. Anschließend werden die Oberflächensegmente, welche den jeweiligen Katalysatorträger umschließen, mit dem neu instanziierten *GroupSurface* Objekt zur Markierung des Trägers im Berechnungsgitter verbunden.

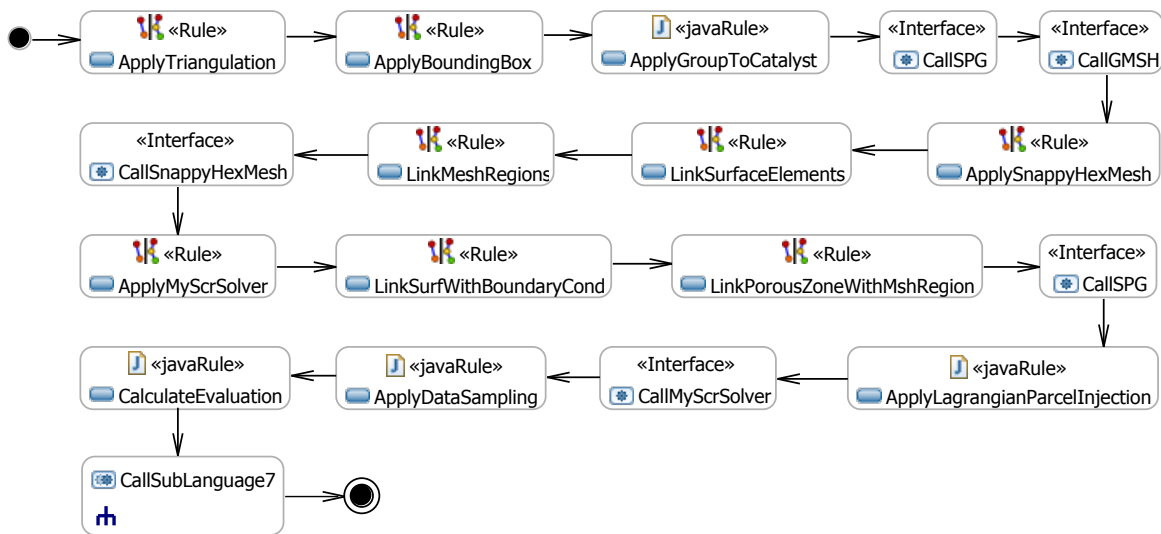


Abbildung 3.21: Vereinfachtes Aktivitätsdiagramm der Teilentwurfssprache 6 zur Vernetzung und CFD-Berechnung des SCR-Systems.

Mit dem nachfolgenden Aufruf des Lösungspfadgenerators und des Vernetzungstools GMSH werden zuerst Daten zwischen den Instanzen ausgetauscht und dann die Vernetzungsoperationen durchgeführt. Außerdem wird die Einhüllende bestimmt und die Gruppierung der triangulierten Oberflächen ausgeführt.

Die drei sich anschließenden Regeln instanziiieren die von *SnappyHexMesh* ererbenden Klassen und verbinden diese mit der triangulierten Geometrie. Sie bringen regelbasiert Netzverfeinerungen auf und verbinden die zu erstellenden Netzregionen mit den gruppierten triangulierten Oberflächen. Nach Ausführung dieser Regeln kann der Vernetzer *SnappyHexMesh* über die Ausführung von *CallSnappyHexMesh* aufgerufen werden.

In den anschließenden Regeln werden die zur Ausführung der CFD-Simulation benötigten Klassen instanziiert und mit den Objekten des Vernetzers verbunden. Dies beinhaltet unter anderem die Initialisierung der Randbedingungen, welche die benötigten Zahlenwerte wie bereits angedeutet aus dem Entwurfsgraph beziehen. Die in der Vernetzung definierte Region des Trägers wird mit der Instanz der Klasse *PorousZone* verbunden, welche die Widerstandskoeffizienten der Träger, die sie aus den Instanzen der Teilentwurfssprache 1 bezieht, enthält. Der Strömungswiderstand der Träger wird über die so definierte poröse Zone in der CFD-Berechnung durch entsprechende Quellterme in den Navier-Stokes-Gleichungen berücksichtigt.

Nach dem erneuten Aufruf des Lösungspfadgenerators zum Austausch und zur Aktualisierung der Daten wird die Klasse *LagrangianParcelInjection* instanziiert und die Spraydaten der HWL-Dosierung aus den Eingabedateien bezogen. Der anschließende Aufruf *CallMyScrSolver* generiert die Eingabedateien zur CFD-Berechnung und lädt die Daten auf den in der Klasse *OpenFOAM* eingetragenen Server zur anschließenden Ausführung hoch. Nach Beendigung der Simulation werden über die Instanziierung und Verbindung der Klasse *DataSampling* mit den entsprechenden Oberflächenelementen die Ergebnisse der Simulation in Form der orts aufgelösten Strömungsdaten aus dem Modell ausgelesen.

Für die abschließende Berechnung des SCR-Wirkungsgrades wird aus der zuvor am SCR-Katalysatoreintritt ausgelesenen Konzentrationsverteilung des Reduktionsmittels das lokale Dosierverhältnis α_{lok} bestimmt. Auf Basis von α -abhängigen Umsatzkennfeldern, die in den Eingabedateien hinterlegt sind, können so die lokalen Stickoxidreduktionen berechnet werden. Diese werden dann, gewichtet mit den lokalen Massenströmen des Abgases, über den Querschnitt aufaddiert und führen so auf eine Abschätzung der sich auf Grund der Reduktionsmittelverteilung einstellenden Stickoxidkonversion des SCR-Systems [196].

Durch diesen kennfeldbasierten Ansatz kann auf eine zeitaufwändige explizite Simulation der chemischen Reaktionen verzichtet werden. Für den Vorentwurf wird dies als ausreichend angesehen. Die dargestellte Teilsprache kann aber durch eine entsprechende Ergänzung des Klassendiagrammes sowie der Regeln auf die Erstellung von CFD-Modellen, welche die Simulation der chemischen Reaktionen und eine Behandlung eines flüssigen Wandfilmes beinhalten, erweitert werden. Nach Abschluss der Berechnung des Wirkungsgrades wird die nachfolgende Teilentwurfssprache 7 zur Strukturberechnung aufgerufen und der Entwurfsgraph übergeben.

3.1.9. Teilentwurfssprache 7: Thermische Strukturberechnung

In der Teilentwurfssprache 7 wird, analog zur in Teilentwurfssprache 6 durchgeführten CFD-Berechnung, die in den vorherigen Schritten generierte Geometrie vernetzt. Anschließend wird ein Simulationsmodell zur Strukturberechnung mit dem Ziel der Bestimmung der thermischen Ausdehnung der Abgasanlage erzeugt. Das Wissen um die thermische Ausdehnung der Abgasanlage liefert wertvolle Ausgangsinformationen für die sich dem Konzeptentwurf anschließende manuelle Detaillierung, zum Beispiel für die Bestimmung der Lagerungspunkte der Anlage.

Die Teilentwurfssprache 7 soll die flexible Wiederverwendung von Entwurfssprachenkomponenten demonstrieren: Dazu wird für die Berechnung der thermischen Ausdehnung der Anlage auf die Funktionalitäten zur Strukturberechnung von OpenFOAM zurückgegriffen. Dabei kann das im vorherigen Abschnitt 3.1.8 dargestellte Plug-in, welches die Grundfunktionalität der Erzeugung von OpenFOAM-Eingabedaten bereits enthält, in einer leicht modifizierten Version wiederverwendet werden.

Die Darstellung soll analog zur vorherigen Teilentwurfssprache 6 in Form einer verkürzten Prinzipdarstellung erfolgen und die wesentlichen Datenstrukturen und Informationsflüsse darstellen. Abbildung 3.22 zeigt das vereinfachte Klassendiagramm der Teilentwurfssprache 7 zur Berechnung der thermischen Ausdehnung der mechanischen Struktur des SCR-Systems.

Die Vernetzung des SCR-Systems findet prinzipiell analog zur Netzerstellung des CFD-Modells des vorherigen Abschnittes 3.1.8 statt. Damit sind die von *OCC_Object* (CAD-Modell), *GMSH* (Vorvernetzung) und *SnappyHexMesh* (Vernetzung des OpenFOAM-Modells) ererbenden Klassen identisch zum Klassendiagramm in Abbildung 3.19 der vorherigen Teilentwurfssprache und werden entsprechend wiederverwendet.

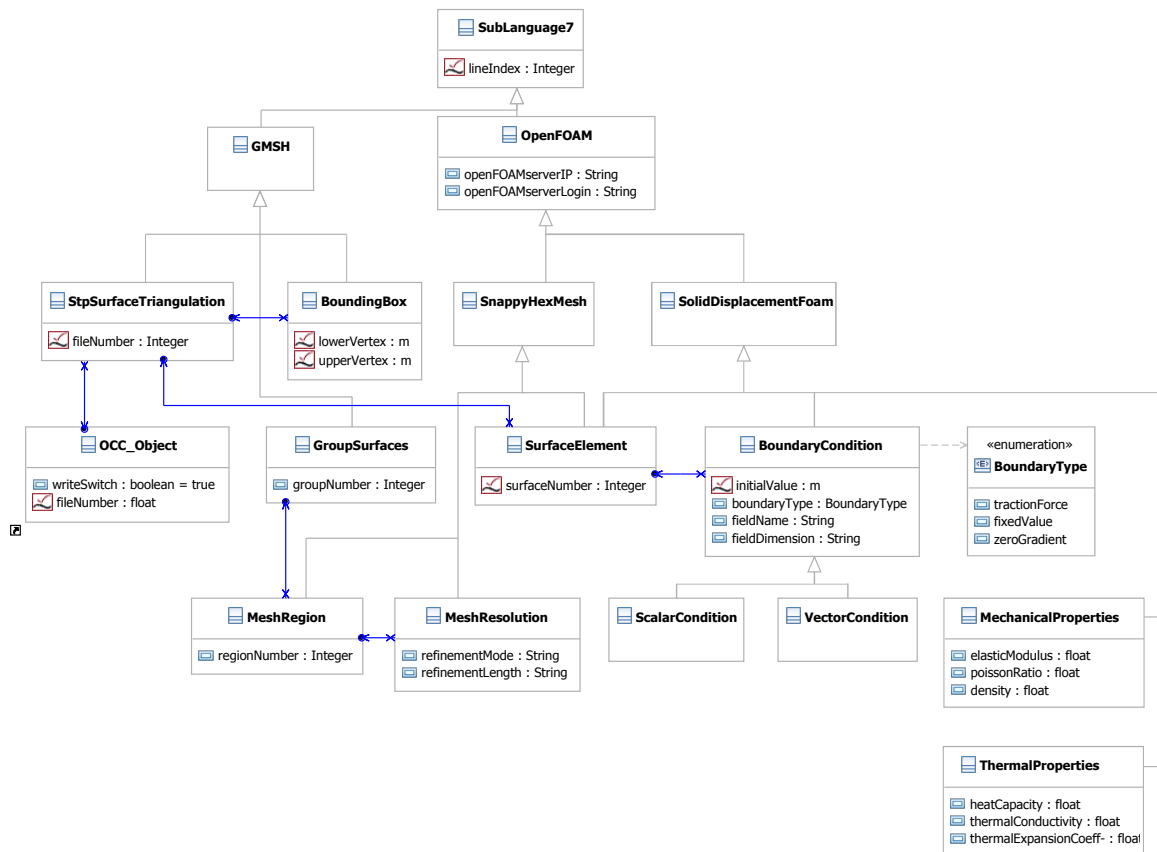


Abbildung 3.22: Vereinfachtes Klassendiagramm der Teilsprache 7 zur Vernetzung und Berechnung der thermischen Ausdehnung des SCR-Systems. Bezeichner der Assoziationen sind aus Gründen der Darstellbarkeit ausgeblendet.

Die von *SolidDisplacementFoam* erbinden Klassen beherbergen die Eingabedaten des verwendeten Strukturmechanik-Solvers *solidDisplacementFoam*. Dieser stationäre Solver berechnet die Verschiebungen des zu berechnenden Modells auf Grund der in *BoundaryType* definierten Randbedingungen (Kräfte, Verschiebungen, Drücke, Temperaturen). Die Zuordnung der Randbedingungstypen zu den Geometrietitäten wird wie im vorherigen Abschnitt mit den Kindern der Klasse *BoundaryCondition* realisiert.

Die Klassen zur Kontrolle der Simulationsdauer (*SimulationConditions*) und zur Auswertung der Simulation (*DataSampling*) aus dem Klassendiagramm des CFD-Plug-ins sind aus Gründen der Einfachheit nicht mehr explizit dargestellt, können aber ohne Änderungen vom CFD-Plug-in übernommen werden. Die neu hinzugekommenen Klassen *MechanicalProperties* und *ThermalProperties* enthalten die mechanischen und thermischen Eigenschaften der Materialien des zu berechnenden SCR-Systemes.

Abbildung 3.23 stellt das Aktivitätsdiagramm der Regelabfolge zur Vernetzung und Strukturberechnung des SCR-Systems dar. Die Generierung der Eingabedaten für die Vernetzung findet analog zur Teilsprache 6 des vorherigen Abschnittes 3.1.8 statt (vergleiche Abbildung 3.21).

Im Gegensatz zur Erstellung des CFD-Modells wird für die grobe Voraussage der thermischen Verformung auf die Berücksichtigung der Katalystorträger in der Erstellung des Strukturberechnungsmodelles verzichtet. Die entsprechenden Regeln und Subaktivitäten aus dem Aktivitätsdiagramm der CFD-Berechnung (Abbildung 3.21) entfallen entsprechend.

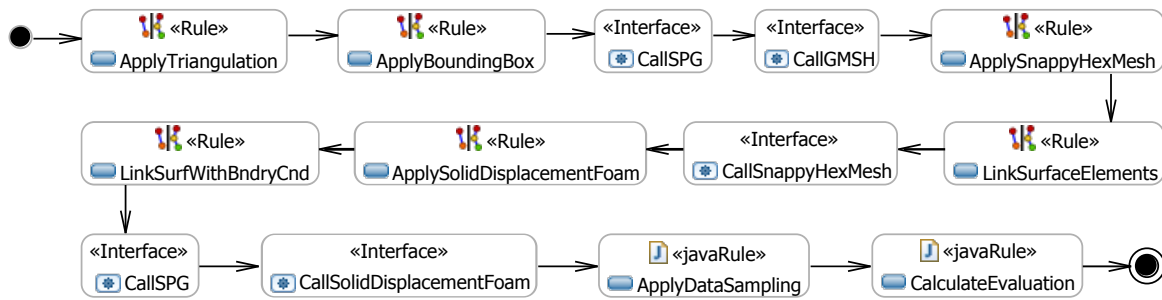


Abbildung 3.23: Vereinfachtes Aktivitätsdiagramm der Teilentwurfssprache 7 zur Vernetzung und Strukturberechnung der thermischen Ausdehnung des SCR-Systems.

Nach dem Aufruf des Vernetzers *CallSnappyHexMesh* werden in Regel *ApplySolidDisplacementFoam* die Klassen zur Speicherung der Eingabedaten für die thermomechanische Berechnung der Anlage instanziiert. Die entsprechenden Randbedingungen werden anschließend über die Erstellung entsprechender Links, zwischen Geometrieobjekten und Objekten zur Speicherung der Eingabe- und Randbedingungen, in Regel *LinkSurfWithBndryCnd* zur Verfügung gestellt.

Dem Aufruf des Lösungspfadgenerators *CallSPG* für den Austausch der für die Berechnung benötigten Daten schließt sich die Erstellung des Berechnungsmodelles sowie die Ausführung des entsprechenden Solvers in *CallSolidDisplacementFoam* an. Über die beiden abschließenden Java-Regeln *ApplyDataSampling* beziehungsweise *CalculateEvaluation* können die Ergebnisse der Berechnung, beispielweise die Position und der Absolutwert der größten Verschiebung, ausgelesen und zurückgegeben werden.

3.2. Anwendungsbeispiele

Dieses Kapitel stellt Anwendungsbeispiele der im vorherigen Abschnitt 3.1 dargestellten Entwurfssprache dar. Dazu werden zwei verschiedene Anwendungsszenarien betrachtet: Im ersten Beispiel *Baumaschine* soll in dem gegebenen Bauraum einer Baumaschine ein SCR-System für einen Motor synthetisiert und berechnet werden. Das zweite Beispiel *Frachtschiff* stellt die Ergebnisse des simultanen Entwurfes von Abgassystemen von vier parallel angeordneten Motoren im Kamin eines hochseetauglichen Schiffes dar. Die beiden Beispiele lehnen sich in ihrer prinzipiellen Schwierigkeit und Gestalt an industrielle Anwendungsprojekte an, werden jedoch aus Datenschutzgründen soweit verändert, dass sie unkenntlich sind.

3.2.1. Baumaschine

Für das Beispiel der Baumaschine wird von einem Motor mit einem Abgasmassenstrom von $\dot{m} = 1270 \frac{\text{kg}}{\text{h}}$ und einer Abgastemperatur von $T_{\text{Abg}} = 400^\circ\text{C}$ bei einer Nennleistung von $P = 250\text{kW}$ ausgegangen. Für die Abgasanlage steht der in Abbildung 3.24 abgebildete Bauraum zur Verfügung. Der Position und Richtung des motorseitigen Abgasauslasses ist rot dargestellt, der Abgasaustritt zur Umgebung ist entsprechend grün gekennzeichnet.

Die Abbildung 3.25 zeigt als Ausschnitt des Entwurfsgraphen das Ergebnis der analytischen Auslegung der Katalysatoren. In den Eingabedateien wurde die Verwendung zylindrischer Träger eingestellt. Als Zielgrößen der Trägersauslegung wurde ein Gegendruck von $p = 2500\text{Pa} = 25\text{mbar}$ und eine Raugeschwindigkeit von $RG = 7.5 \frac{1}{\text{s}} = 27000 \frac{1}{\text{h}}$ definiert (Instanz *catDimension1*). Auf Basis dieser Zielgrößen wurden vier serielle Standardträger mit Durchmesser

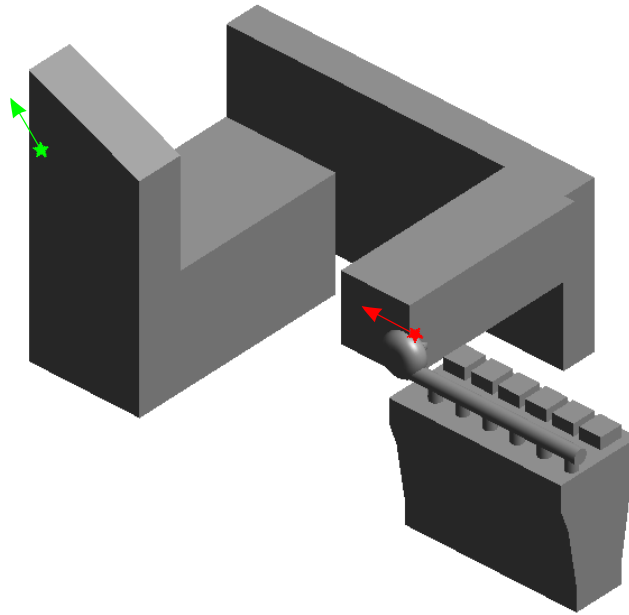


Abbildung 3.24: Für den Entwurf des SCR-Systems zur Verfügung stehender Bauraum des Beispiels Baumaschine.

$d = 342mm$ und Einzellängen von $l = 100mm$ ausgewählt (Instanz *circCatalyst2* und *singleCylindricalBrickData*). Die Instanz *engine0* enthält die für die Auslegung benötigten Motordaten.

Es sollen beispielhaft zwei topologische Variationen der Mischstrecke entworfen werden. Die erste Variante bestehe aus einer geraden Rohrstrecke, an deren Eintritt das Reduktionsmittel eindosiert werden soll. Die zweite Variante der Mischstrecke enthalte einen Hydrolysekatalysator mit Ein- und Ausströmungskonen auf den das Reduktionsmittel dosiert wird. Die Abbildung 3.26 stellt die Geometrie der synthetisierten Komponenten im Bauraum nach erfolgter Ausführung der Subsprache 4 (analytische Auslegung, Packaging und Geometrieerstellung der Komponenten) dar. Die einhüllenden Quader der Komponenten, welche während des Packagings der Komponenten vom Bauraum abgezogen werden, sind neben dem gefilterten Punktpfad der Platzierung der Komponente aus der Packaging-Teilsprache 2 (Abschnitt 3.1.4) dargestellt.

Unter Verwendung der in Teilsprache 4 bestimmten Einbaupositionen wird mit Hilfe der Verrohrungssprache aus Kapitel 4.1.1, bei manuell überwachter Ausführung des Produktionssystems mit parametrischer Optimierung und Reinitialisierung (siehe dazu Kapitel 4.1.5), eine Verrohrung aus Standardrohrbögen generiert (Bogenwinkel $\bar{\alpha} = 30^\circ, 45^\circ, 60^\circ, 90^\circ$). Diese besitzen mit dem Rohrdurchmesser identische Biegeradien $R = D = 0.15m$. Abbildung 3.27 stellt das CAD-Modell des fertig verrohrten SCR-Systems zum Abschluss von Teilentwurfssprache 5 dar.

Das so generierte CAD Modell wird für die CFD-Berechnung des SCR-Systems verwendet. Abbildung 3.28 zeigt das mit *snappyHexMesh* vernetzte Modell des SCR-Systems mit grün dargestellten Katalysatorträgern. Die Laufzeit der Vernetzung liegt auf dem verwendeten Linux-Server¹⁹ im Bereich von Minuten. Das vernetzte CFD-Modell weist eine Zellzahl von ungefähr 250000 Elementen auf. Die charakteristische Zellgröße der Vernetzung wird passend zum Turbulenzmodell gewählt. In dem gezeigten Beispiel wird ein High-Re-k-epsilon-Turbulenzmodell gewählt, welches die wandnahen Bereiche des Geschwindigkeitsprofils der Strömung mit einer

¹⁹Linux-Server: 8 GB Arbeitsspeicher und 3 GHz

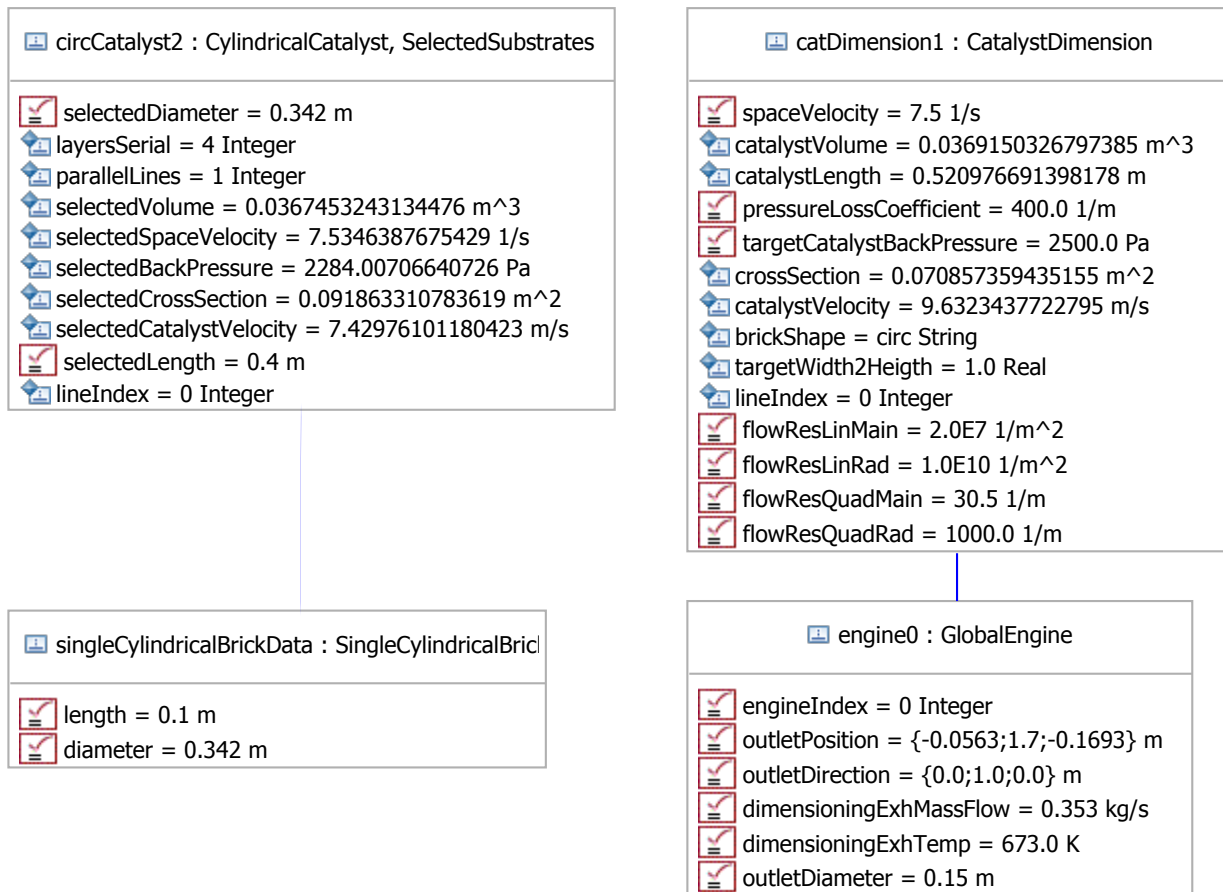


Abbildung 3.25: Ausschnitt des Entwurfsgraphen (Instanzendiagramm): Ergebnis der Analytischen Auslegung der Katalysatoren des Beispiels Baumaschine. Die Beschriftungen der Verbindungen sind ausgeblendet.

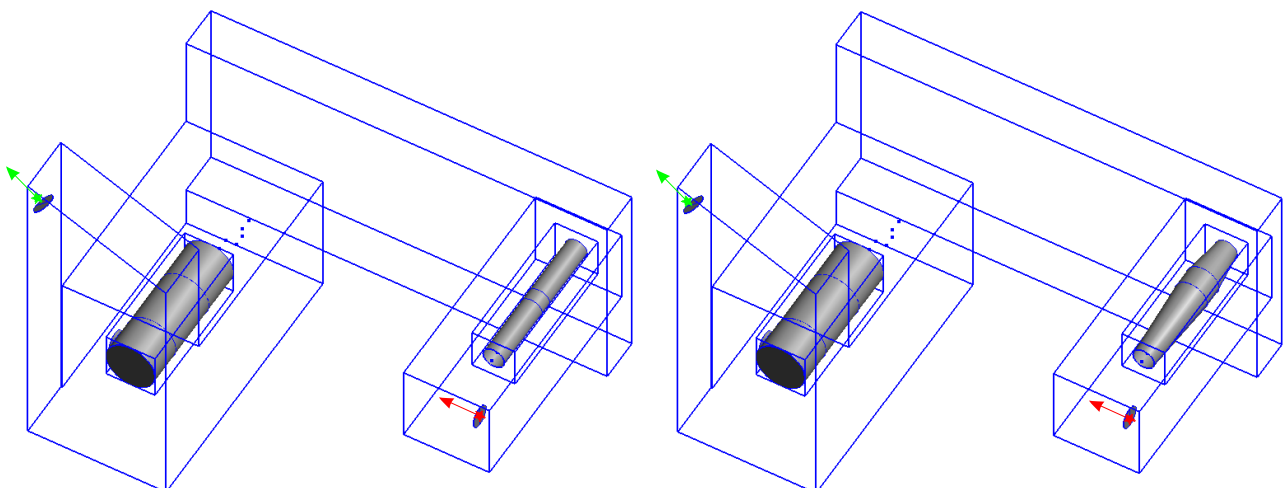


Abbildung 3.26: Packaging der Komponenten des Beispiels Baumaschine mit einhüllenden Quadern, welche vom Bauraum abgezogen werden. Linke Seite: Gerades Rohr als Mischstrecke. Rechte Seite: Hydrolysekatalysator in der Mischstrecke.

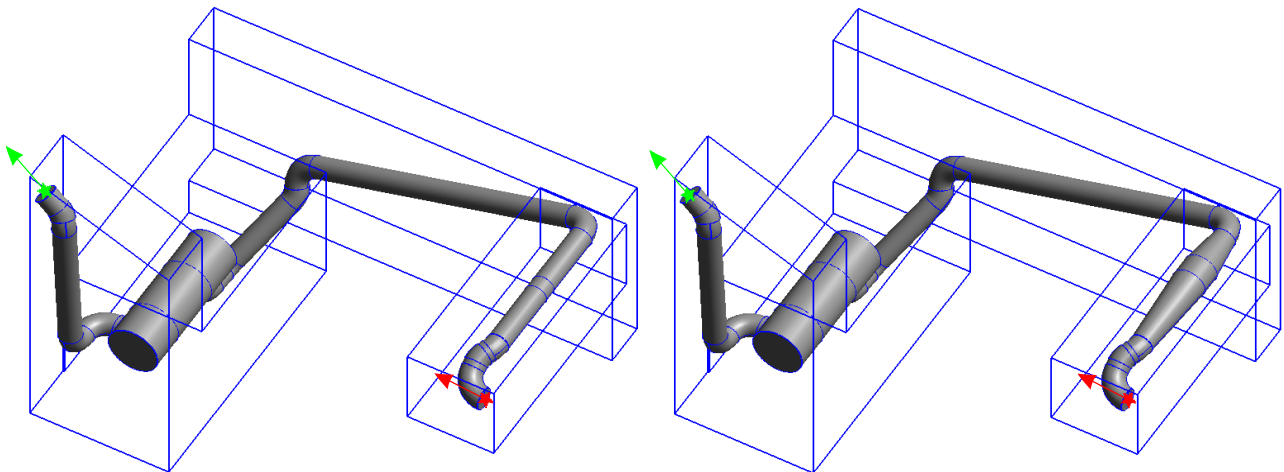


Abbildung 3.27: CAD-Modell der SCR-Systeme des Beispiels Baumschine. Linke Seite: Gerades Rohr als Mischstrecke. Rechte Seite: Hydrolysekatalysator in der Mischstrecke.

Wandfunktion approximiert. Die Dicke der wandnächsten Elementschicht lässt sich aus den Werten des dimensionslosen Wandabstandes y^+ abschätzen [31, 162].

Die eingefärbte Oberfläche des SCR-Systems in Abbildung 3.29 stellt den Verlauf des Wertes des totalen Drucks über das Abgassystem dar. Der Abschnitt der Eindosierung des Reduktionsmittels auf den Hydrolysekatalysator ist durchsichtig dargestellt, so dass die Tropfen-Parzellen der CFD-Berechnung zu erkennen sind. Diese sind entsprechend dem Tropfendurchmesser der jeweiligen Parzelle farbcodiert. Im Beispiel wird ein typischer Off-Highway HWL-Injektor mit einem Spraykonuswinkel von $\alpha_{inj} = 30^\circ$, einem mittleren Sauterdurchmesser von $SMD = 120\mu m$ und einer Tropfengeschwindigkeit von $v_{inj} = 25m/s$ verwendet.

Die im automatisierten Postprocessing an ausgewählten Querschnitten berechneten Werte des totalen Druckes sowie der Druckverluste der beiden Entwürfe (mit und ohne Hydrolysekatalysator) sind in der Tabelle 2 dargestellt. Die Druckverlustwerte des Entwurfes mit Hydrolysekatalysator sind, auf Grund des zusätzlichen Gegendruckes des Hydrolyse-Katalysatorträgers, geringfügig höher.

Der Bereich des größten Druckabfalls im Abgassystem ist über die SCR-Katalysatorträger zu erkennen. Die Gegendrücke des SCR-Katalysators liegen geringfügig unter der in der analytischen Auslegung angepeilten Obergrenze des Druckverlustes, dargestellt im Instanzendiagramm in Abbildung 3.25.

Die Abbildung 3.30 stellt den Verlauf der Ammoniak-Konzentration über das SCR-System wiederum an ausgewählten Querschnitten des Entwurfes mit Hydrolysekatalysator dar. Die zunehmend gleichmäßigere Vermischung des Ammoniaks mit dem Abgas in der langen Rohrstrecke zwischen Hydrolysekatalysator und SCR-Katalysatorbox ist gut zu erkennen und erreicht einen Gleichverteilungskoeffizienten von $UI \approx 0.95$ am Eintritt des SCR-Katalysator.

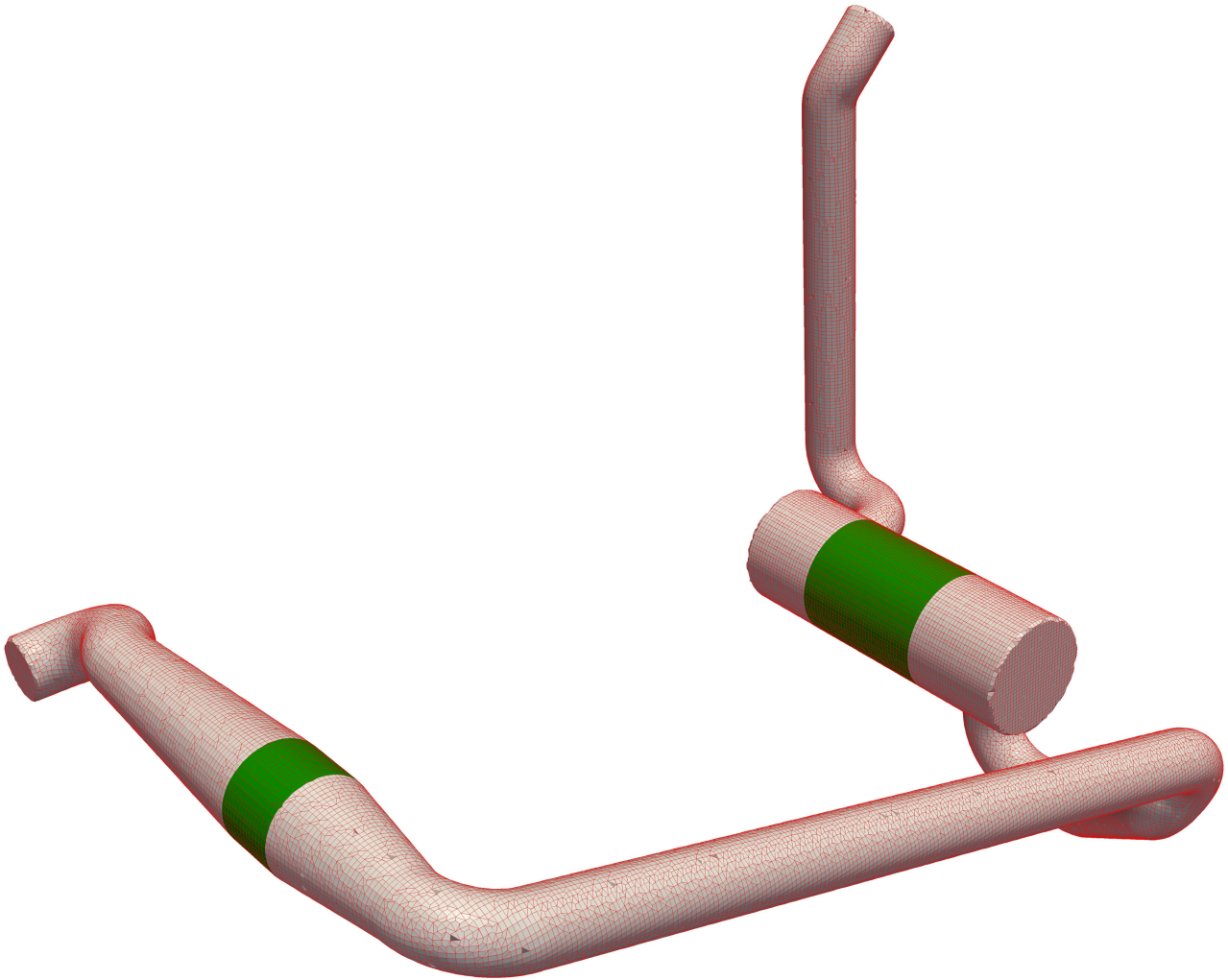


Abbildung 3.28: Für die CFD-Berechnung mit dem OpenFOAM-Vernetzter *snappyHexMesh* erstelltes Berechnungsmodell des SCR-Systems mit Hydrolysekatlaysator. Die Netzregionen der Katalysatorträger sind grün dargestellt.

Position	Tot. Druck ohne Hyd.-Kat.	Tot. Druck mit Hyd.-Kat.
Motorauslass	104247 Pa	104264 Pa
Eintritt Mischstrecke	104138 Pa	104163 Pa
Austritt Mischstrecke	104074 Pa	103883 Pa
Eintritt SCR-Box	103716 Pa	103533 Pa
Eintritt SCR-Kat	103434 Pa	103184 Pa
Austritt SCR-Kat	101111 Pa	101026 Pa
Auspuffende	100663 Pa	100556 Pa
Differenz	3584 Pa	3708 Pa

Tabelle 2: Druckverluste des Beispiels Baumaschine mit und ohne den Einsatz eines Hydrolysekatlaysators in der Mischstrecke.

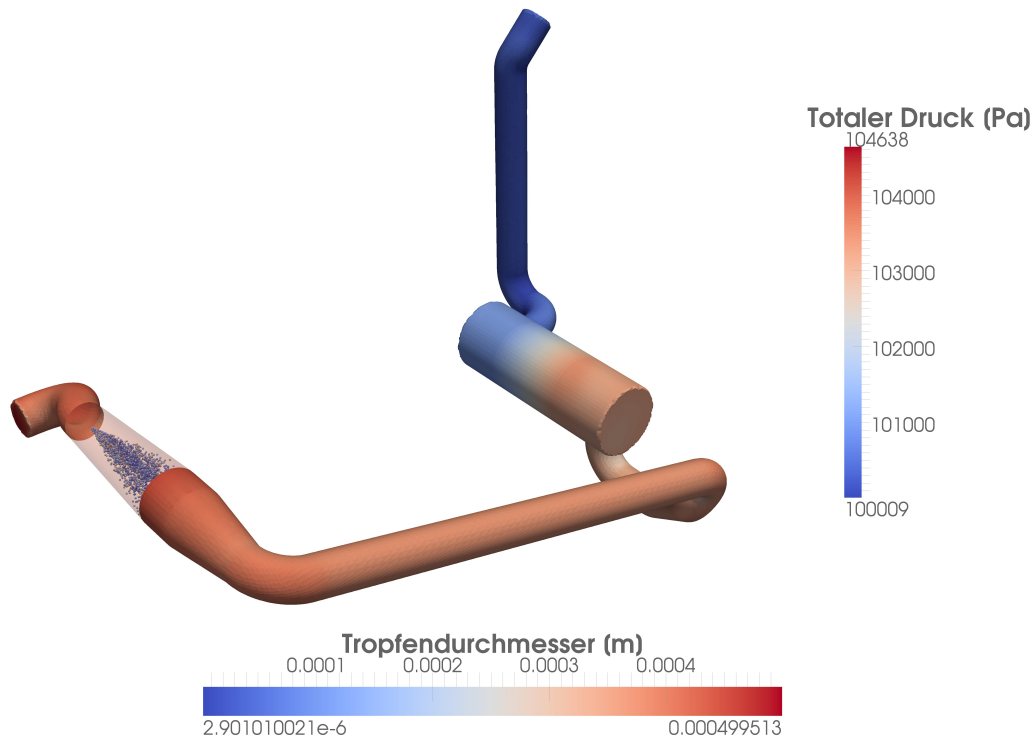


Abbildung 3.29: CFD Ergebnisse des Beispiels Baumaschine. Färbung der Oberfläche des SCR-Systems: Totaler Druck. Die Tropfen des Reduktionssmittels sind entsprechend dem Tropfendurchmesser eingefärbt.

Abbildung 3.31 zeigt die nach Abschnitt 3.1.8 geschätzten NO_x-Konversionen des SCR-Systems für verschiedene Katalysatoreffizienzen ($\eta = 100\%$, 90% , 80%). Dabei wurde die mit CFD berechneten Ammoniakkonzentrationen und Verteilungen des Abgasmassenstroms am Eintritt des SCR-Katalysators berücksichtigt.

Der Umsatz eines Katalysators skaliert qualitativ mit dem Gleichverteilungskoeffizient und dient deshalb in der Praxis als schnelle Bewertungs- und Vergleichsgröße der Strömungszuführung verschiedener Katalysatoreinbauten. In der praktischen Auslegung von Abgassystemen werden im Allgemeinen Gleichverteilungswerte $UI \geq 0.95$ gefordert. Solcherlei Gleichverteilungswerte führen, als grober Richtwert aus der Praxis, bei entsprechender Dosiermenge auf Stickoxidumsätze im Bereich von 90% ohne nennenswerten Ammoniak schlupf durch lokale Überkonzentrationen des Reduktionsmittels. Für einen idealen Katalysator ($\eta = 100\%$) und ein stöchiometrisches Dosierverhältnis von $\alpha = 1.0$ ergibt sich in der praktischen Anwendung eine gute Übereinstimmung der Konversionsleistung mit dem Gleichverteilungskoeffizienten.

Der Vergleich der beiden Entwürfe zeigt geringfügige Nachteile im Wirkungsgrad bei Verwendung eines Hydrolysekatalysators. Dies wird wahrscheinlich von der Verminderung der Turbulenz stromabwärts des Hydrolysekatalysators verursacht. Die Strömung in den Kanälen des Hydrolysekatalysators erfährt eine Reduktion der Turbulenzintensität, so dass der Grad der Turbulenz stromabwärts der Katalysatorträger stark vermindert ist. Dies resultiert mutmaßlich in einer schlechteren Vermischung des Ammoniaks über den Querschnitt des Abgases durch reduzierte turbulente Diffusion.

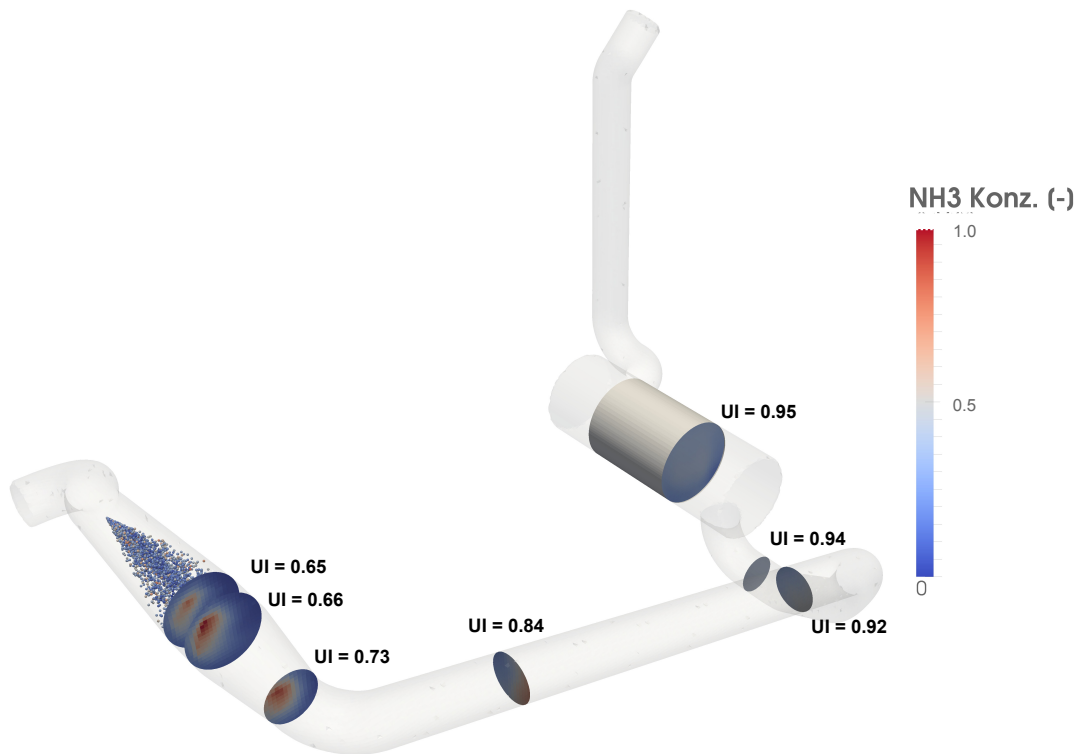


Abbildung 3.30: Normierte Ammoniakkonzentration an verschiedenen Querschnitten zwischen Eintritt des Hydrolysekatalysators und Eintritt SCR-Katalysator inklusive zugeordneter Gleichverteilungswerte (UI).

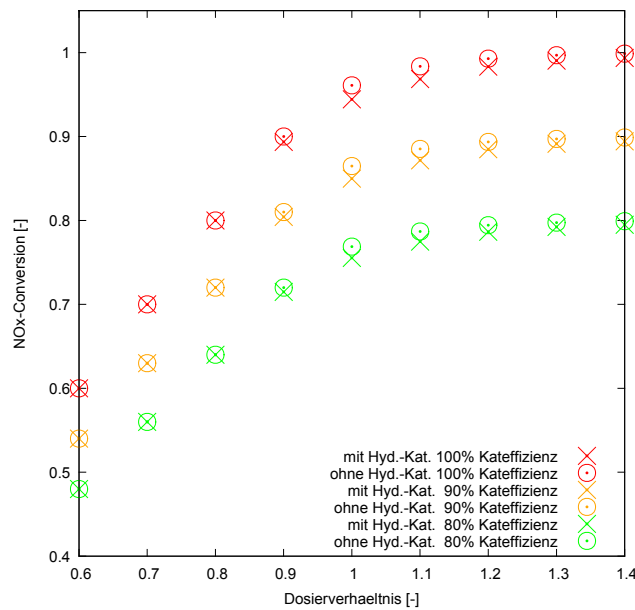


Abbildung 3.31: Nach Kapitel 3.1.8 abgeschätzter SCR-Wirkungsgrad für verschiedene Katalysatoreffizienzen für die SCR-Systeme mit und ohne Hydrolysekatalysator.

Die Verwendung des Hydrolysekatalysators weist aber einen entscheidenden Vorteil bezüglich der Gefahr der Ablagerungsbildung auf: Durch die gezielte Eindosierung des Reduktionsmittels auf den beschichteten Hydrolysekatalysator kann im Allgemeinen auch bei niedrigen Temperaturen eine rückstandsfreie Verdampfung des Reduktionsmittels gewährleistet werden [166]. Somit sind nur noch die Anteile des Reduktionsmittels, welche auf die Wände der Verrohrung verweht werden, kritisch bezüglich der Bildung von Ablagerungen.

Vergleicht man die im regelbasierten Postprocessing des Systems bestimmten Massennanteile des auf die Wände der Anlage treffenden Reduktionsmittels, so ergibt sich für den Entwurf mit Verwendung eines Reduktionskatalysators ein erheblicher Vorteil: Bei Verwendung eines Hydrolysekatalysators treffen nur 3% der eingespritzten Reduktionsmittelmasse auf die Wände des Systems. Beim Verzicht auf einen Hydrolysekatalysator landen ungefähr 80% des Reduktionsmittels auf Wänden. Daraus folgt eine erhebliche Erhöhung der Gefahr der Ablagerungsbildung beim System ohne Hydrolysekatalysator. Die automatisiert generierte Bewertungsgröße des auf die Wand treffenden Massenanteils des eindosierten Reduktionsmittels steht damit, bei der Verwendung der dargestellten Entwurfssprache, schon in einem frühen Konzeptstadium zur Bewertung des SCR-Systems zur Verfügung.

Abbildung 3.32 zeigt die thermische Ausdehnung des SCR-Systems im vorliegenden Beispiel, welche mit der in Subsprache 7 aus Kapitel 3.1.9 automatisiert generierten und ausgeführten stationären Strukturanalyse berechnet wurde. Die Verschiebung des SCR-Systems am Abgasauslass des Motors wurde auf null gesetzt. Die Abgastemperatur von $T_{Abg} = 400^{\circ}C$ wurde auf das gesamte Modell aufgebracht. Der unverschobene Zustand entspricht der Ausdehnung der Anlage bei Standardbedingungen.

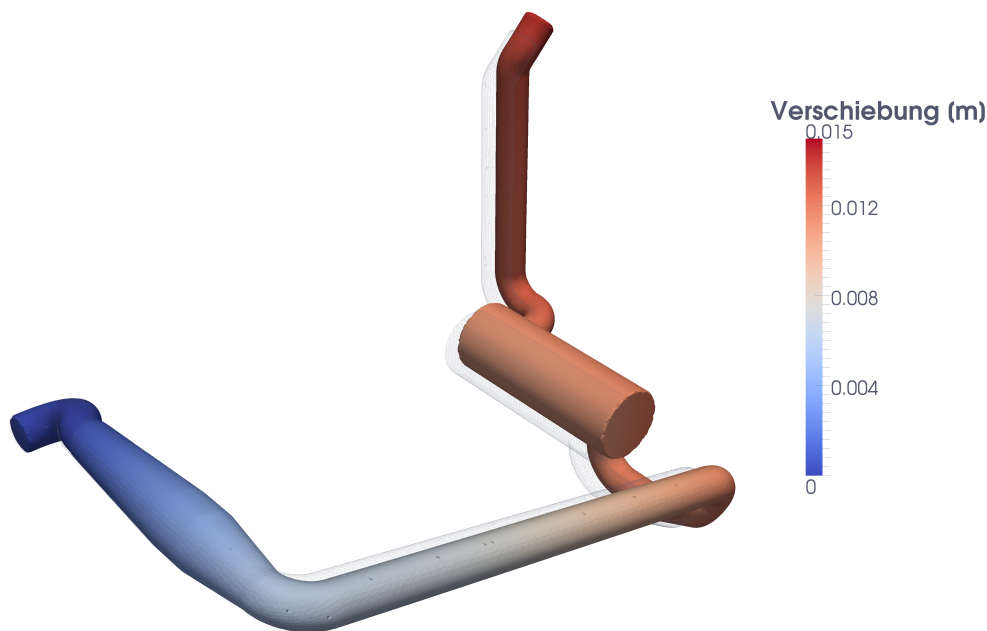


Abbildung 3.32: Strukturberechnung: Thermische Ausdehnung des Abgassystems bei einer Temperatur von $T_{Abg} = 400^{\circ}C$. Die visuell dargestellte Verschiebung relativ zum unverformten Drahtgittermodell ist um den Faktor zehn, bezogen auf die farbcodierten tatsächlichen Verschiebungswerte, überhöht dargestellt.

Im Ergebnis der thermomechanischen Berechnung ergibt sich eine Ausdehnung des SCR-Systems, durch die Erwärmung der Struktur auf Grund des durchströmenden heißen

Abgases, von ungefähr 1.5cm. Diese konzeptionelle Berechnung der absoluten Ausdehnung liefert wertvolle Informationen, beispielsweise bei der Bestimmung der Lagerungspositionen und der elastischen Lagerelemente, in der späteren Ausdetaillierung des SCR-Systementwurfes.

Die Tabelle 3 zeigt eine Auflistung der Laufzeiten der einzelnen Syntheseschritte der Entwurfssprache des vorliegenden Beispiels der Baumaschine. Die CFD-Berechnung umfasst eine Strömungsdauer von $t = 0.5s$, während dessen das Reduktionsmittel kontinuierlich eindosiert wird. Zuvor wurde das Strömungsfeld mit den Ergebnissen einer ebenfalls vollautomatisiert erzeugten und durchgeführten stationären Strömungssimulation, ohne Eindosierung, initialisiert.

Syntheseschritt	Laufzeit	relative Laufzeit
Randbedingungen einlesen	< 1 min	0.1%
Analytische Auslegung	< 1 min	0.1%
Packaging und Synthese CAD-Komponenten	3 min	0.3%
Routing und Synthese CAD-Rohrstrecken (manuell)	20 min	2.1%
Vernetzung des Modells	10 min	1.0%
CFD-Berechnung ($t = 0.5s$)	900 min	93%
Strukturberechnung	30 min	3.1%
Auswertung und Postprocessing	5 min	0.5%
Gesamt	970 min	100%

Tabelle 3: Laufzeiten der Entwurfssprache des Beispiels Baumaschine auf einem PC mit 2GB RAM und 2 CPUs à 2 GHz. CFD-Berechnung auf Linux Server mit 8 GB RAM und Verteilung auf 4 CPUs à 3 GHz.

Für das durchgeführte Beispiel zeigt sich, dass die gesamte Modellerstellung inklusive der Vernetzung und der Auswertung der Ergebnisse nur einen Bruchteil der Laufzeit der CFD-Berechnung ausmacht. Die Dauer der CFD-Simulation beläuft sich auf der zur Verfügung stehenden Hardware auf circa einen halben Tag. Die CFD-Berechnungen lassen sich prinzipiell auch auf Höchstleistungsrechnern bzw. Clustern ausführen, womit die Berechnungsdauer reduziert werden kann oder mehrere Konfigurationen gleichzeitig berechnet werden können. Damit ist eine vollautomatische Optimierung von SCR-Systemen, wie beispielsweise im Bottom-Up-Entwurf von Strommasten mit Hilfe von Entwurfssprachen in [2] dargestellt, mit zukünftig steigenden Rechenkapazitäten in unmittelbarer Reichweite. Dies soll aber nicht mehr Teil dieser Arbeit sein, da sich mit der zur Verfügung stehenden Rechenleistung in einem sinnvollen Zeitrahmen keine belastbaren Optimierungsergebnisse erzielen lassen. Der erfolgreiche Prinzipnachweis der vollautomatisierten Generierung von Systemvarianten soll in diesem Rahmen daher genügen.

3.2.2. Frachtschiff

In diesem Abschnitt wird das Ergebnis des Entwurfes von vier parallelen Abgassystemen für vier Motoren im Abgaskamin eines Hochseefrachters mit der in Kapitel 3.1 beschriebenen Entwurfssprache dargestellt. Es sollen nur die wichtigsten Ergebnisse in Kürze dargestellt werden, da die verwendeten Teilsprachen identisch mit denen des vorherigen Beispiels der Baumaschine sind und so die prinzipielle Übertragbarkeit der gezeigten Entwurfssprache für unterschiedliche Fahrzeuganwendungen demonstriert werden soll. Das Beispiel Frachtschiff soll primär zur Überprüfung und Darstellung der Funktionalitäten zur simultanen Synthetisierung paralleler Abgassysteme in einem gemeinsamen Bauraum dienen.

Im vorliegenden Beispiel Frachtschiff wird von jeweils vier Motoren mit einem Abgasmassenstrom von jeweils $\dot{m} = 10000 \frac{kg}{h}$ und einer Abgastemperatur von $T_{Abg} = 325^\circ C$ bei einer Nennleistung von $P = 1300 kW$ ausgegangen. Der für das Abgassystem zur Verfügung stehende Teil des Abgaskamines des Schiffes ist in Abbildung 3.33 dargestellt. Die Positionen und Richtungen der Abgasauslässe der Motoren sind rot gekennzeichnet, die Abgasübergabe zur Umgebung am Kaminende ist jeweils grün dargestellt.

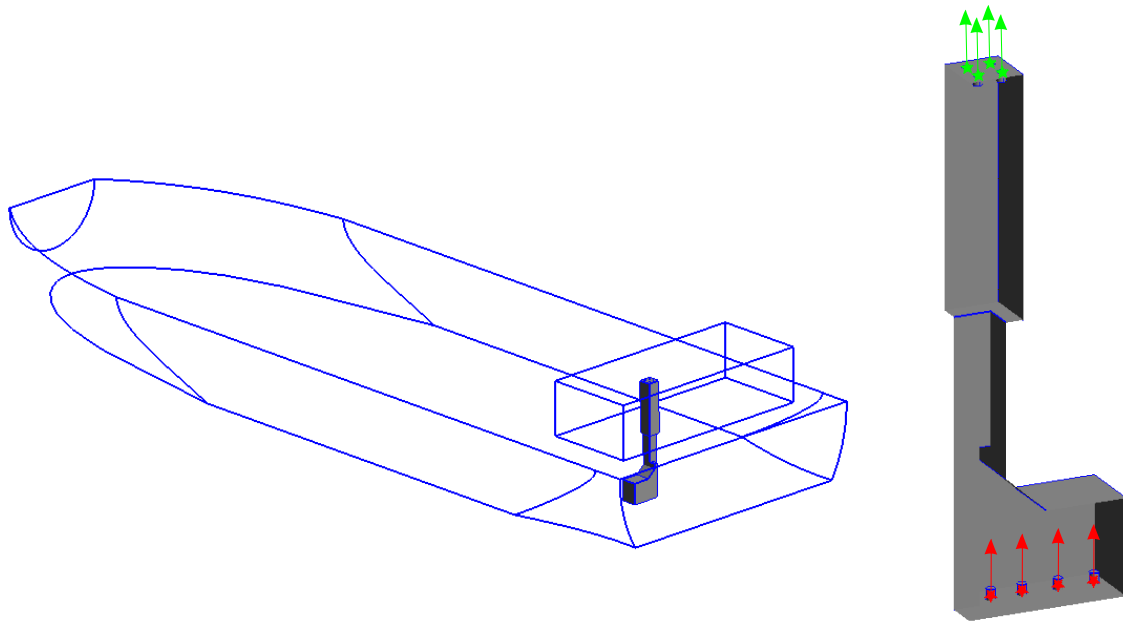


Abbildung 3.33: Für den Entwurf der vier SCR-Systeme zur Verfügung stehender Bauraum des Beispiels Frachtschiff. Linke Seite: Bauraum innerhalb des Schiffsrumpfes eines Hochseeschiffes. Rechte Seite: Bauraum im Detail (rot: Abgasaustritte des Motors, grün: Abgasaustritt Kaminende).

Für Anwendungen in Hochseeschiffen wird im Allgemeinen auf robuste, mit der Verfahrens- und Anlagentechnik verwandte Technik zurückgegriffen. Dies bedeutet die Verwendung rechteckiger Katalysatorträger, welche axial an- und abgeströmt werden. In der Regel wird ein Seitenverhältnis (Trägerbreite zu Trägerhöhe) von eins angestrebt, was auf SCR-Reaktoren mit quadratischem Querschnitt führt.

Auf Basis dieser Eingabedaten wird in Teilsprache 1 die analytische Auslegung der vier identischen SCR-Katalysatorträger berechnet. Dabei wird eine für diese Anwendung typische, niedrige Soll-Raumgeschwindigkeit von $RG = 8000 \frac{1}{h}$ angesetzt. Der maximale Auslegungsgegendruck der Träger wird auf $p = 1500 Pa = 15 mbar$ festgesetzt. Die Wahl der niedrigen Raumgeschwindigkeit führt nach Gleichung (3.1) auf ein großes Katalysatorvolumen. In Kombination mit dem niedrigen Zielgegendruck trägt diese Auslegung sowohl dem üppigen Bauraumangebot in der Großschiffanwendung, als auch der Forderung nach geringen Kraftstoffverbräuchen und der Notwendigkeit von Alterungsreserven Rechnung. Der zu verwendende Rohrdurchmesser liegt für diese Anwendung bei $D_{rohr} = 0.5 m$.

Diese Eingangsgrößen führen dann, unter Verwendung von marktüblichen Trägerelementen mit Abmaßen von $b \times h = 150 mm \cdot 150 mm$ und einer Länge von $l = 400 mm$, in der analytischen Auslegung auf eine Konfiguration von drei Trägerebenen mit jeweils $N = 6 \cdot 6 = 36$ Trägerelementen pro Ebene, für jeden der vier Motoren. Dies entspricht einer Ist-Raumgeschwindigkeit von $RG_{ist} = 8100 \frac{1}{h}$.

Zur Minimierung des Druckverlustes des SCR-Systems wird auf die Verwendung eines Hydrolysekatalysators verzichtet. Die Mischstrecken sollen aus geraden Rohrstrecken bestehen. Die Mischstrecken und Katalysatorboxen sollen jeweils bevorzugt in einer horizontalen Ebene angeordnet sein. Dies kann über ein entsprechendes Attribut, siehe Abschnitt 3.1.4, erzwungen werden. Für die relative Positionierung der Komponenten zueinander wird eine 2×2 -Anordnung in den Eingabedaten gewählt. Die Mischstrecken, SCR-Katalysatorboxen und die Abgasauslässe am Kaminende jeder Abgaslinie sind jeweils übereinander angeordnet, so dass es keine Kreuzungen in den Rohrstreckenabschnitten stromabwärts der Mischstrecke geben soll.

In den nachfolgenden Teilsprachen 3 und 4 wird auf Basis der analytischen Auslegung der Katalysatorträger sowie der Eingabedaten an den zuvor bestimmten Einbaupositionen das CAD-Modell der beiden Komponenten Mischstrecke und SCR-Katalysatorbox generiert. Abbildung 3.34 zeigt auf der linken Seite das entsprechende CAD-Modell der Komponenten.

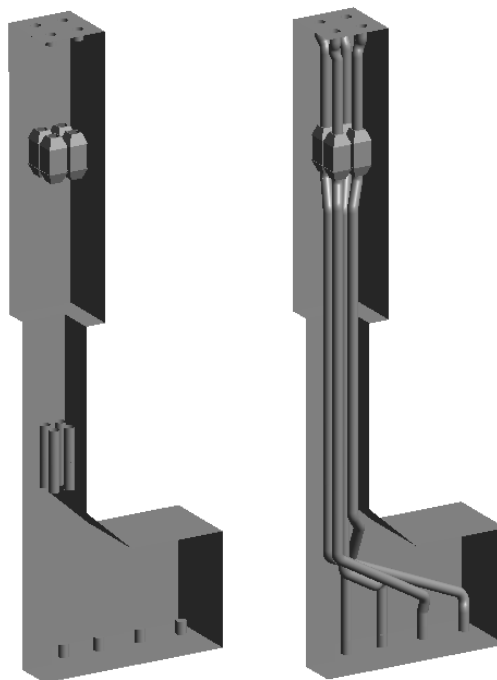


Abbildung 3.34: Linke Seite: CAD-Modell der Komponenten Mischstrecken und SCR-Katalysatorboxen des Beispiels Frachtschiff. Rechte Seite: CAD-Modell der Komponenten inklusive der in Kapitel 4.2.1 bestimmten Rohrstrecken.

Die Verrohrung des Abgassystems unter Verwendung von standardisierten Rohrbögen soll in diesem Abschnitt nicht explizit dargestellt werden. Die Rohrstrecken werden mit der in Kapitel 4.1 ausführlich dargestellten Verrohrungssprache generiert. Die Anwendung der Verrohrungssprache zur Synthese der Rohrstrecken des vorliegenden Beispiels ist in Kapitel 4.2.1 unter Verwendung des Produktionssystems aus Kapitel 4.1.5 ausführlich beschrieben. Die Abbildung 3.34 stellt auf der rechten Seite das Ergebnis der vollautomatisierten Synthese der Rohrstrecken mit Hilfe des Optimierungsalgorithmus der Simulierten Abkühlung dar.

Die Vernetzung des CAD-Modells für die nachfolgende CFD-Berechnung wird wiederum unter Verwendung eines High-Re-k-epsilon-Turbulenzmodell ohne Verfeinerung der Wandschicht durchgeführt. Auf Grund der vergleichbaren Abgasgeschwindigkeiten, Dichten und dimensionslosen Wandabstände y^+ [31, 162] ergeben sich für beide Beispiele vergleichbare Zellgrößen.

Verglichen mit dem Beispiel Baumaschine führt dies, auf Grund der größeren geometrischen Abmaße, auf eine sehr viel größere Elementzahl des CFD-Rechengitters im Beispiel Frachtschiff. Dies schlägt sich vor allem in den Laufzeiten der CFD-Berechnung des Entwurfes der vier parallelen Abgaslinien nieder. Die entsprechenden Laufzeiten für die Synthese und Berechnung der vier Systeme sind in der Tabelle 4 dargestellt.

Syntheseschritt	Laufzeit	relative Laufzeit
Randbedingungen einlesen	< 1 min	0.002 %
Analytische Auslegung	< 1 min	0.002 %
Packaging und CAD Komponenten	10 min	0.02 %
Routing und CAD Rohrstrecken (Sim. Abköhl.)	1800 min	3.4 %
Vernetzung der Modelle	360 min	0.69 %
CFD-Berechnungen ($t = 1s$)	50000 min	95.8 %
Auswertung und Postprocessing	20 min	0.04 %
Gesamt	52192 min \approx 36 Tage	100.0 %

Tabelle 4: Laufzeiten der Entwurfssprache des Beispiels Frachtschiff auf einem PC mit 2GB RAM und 2 CPUs à 2 GHz. CFD-Berechnung auf Linux Server mit 8 GB RAM und 4 CPUs à 3 GHz.

Die simulierte Strömungsdauer $t = 1s$ muss für das Beispiel Frachtschiff größer als im Beispiel Baumaschine gewählt werden, da bei vergleichbaren Strömungsgeschwindigkeiten die Transportstrecke des Reduktionsmittels, von der Eindosierung in der Mischstrecke zum SCR-Katalysator, wesentlich größer ist. Zusammen mit der schon erwähnten größeren Elementzahl ergeben sich die entsprechend extrem höheren Laufzeiten.

Die Abbildung 3.35 zeigt die geschätzte NO_x-Konversionskurve als Ergebnis der CFD-Berechnung einer Abgaslinie des Beispiels Frachtschiff analog zur Darstellung des Beispiels Baumaschine. Eine der zentralen Fragestellungen des dem Beispiel zugrundeliegenden Projektes war die Übertragbarkeit des im Beispiel Baumaschine verwendeten Injektors auf den Einsatz in der betrachteten Frachtschiffsanwendung. Aus diesem Grund wird im Beispiel Frachtschiff ein Injektor mit denselben Tropfen-Eigenschaften wie im Beispiel Baumaschine eingesetzt.

Der Vergleich der Konversionskurve mit dem Beispiel Baumaschine in Abbildung 3.31 zeigt wesentlich schlechtere Konversionswerte für das Beispiel Frachtschiff. Dies ist im Wesentlichen auf zwei Gründe zurückzuführen:

- Weniger Rohrbögen, beziehungsweise Biegungen in der Strecke zwischen Eindosierung (Mischstrecke) und SCR-Katalysatorbox, führen auf geringere Turbulenzwerte und somit schlechtere Durchmischung des Reduktionsmittels mit dem Abgasstrom.
- Der verwendete Injektor hat einen zu geringen Spraykegel beziehungsweise einen zu niedrigen Tropfenimpuls, so dass bei der Einspritzung keine ausreichende Querverteilung der Tropfen über den im Beispiel Frachtschiff größeren Rohrquerschnitt erreicht wird.

Die oben genannte Fragestellung der Übertragbarkeit und Anwendbarkeit des im Beispiel Baumaschine verwendeten Injektors, konnte in dem zu Grunde liegenden Projekt durch Verwendung der vorliegenden Entwurfssprache negativ beschieden werden. Dies führte im weiteren Projektverlauf auf die zusätzliche Verwendung eines Mischers stromabwärts des Injektors sowie auf die Auswahl eines alternativen HWL-Injektors mit größerem Tropfenimpuls um den Rohrquerschnitt besser mit dem HWL-Spray zu überdecken.

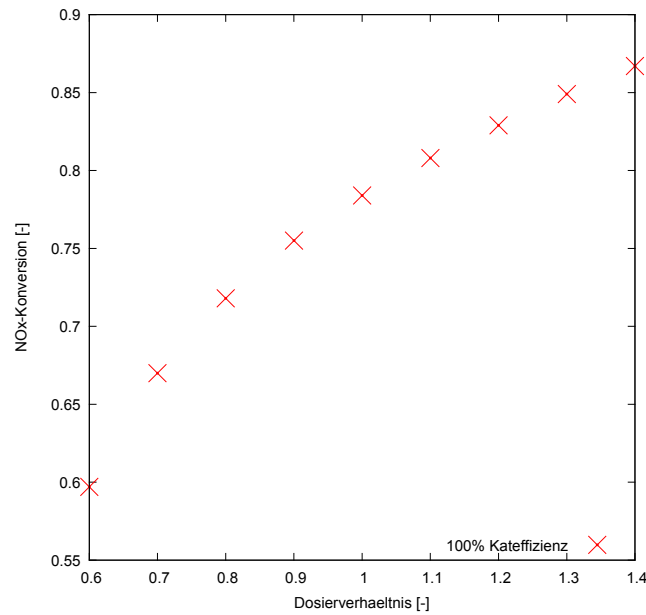


Abbildung 3.35: Nach Kapitel 3.1.8 abgeschätzter SCR-Wirkungsgrad eines idealen Katalysators ($\eta = 100\%$) für ein SCR-System des Beispiels Frachtschiff.

Als Assistenzsystem zur automatisierten Generierung und Auswertung von SCR-Systementwürfen bietet die beschriebene Entwurfssprache eine erhebliche Effizienzsteigerung und Qualitätssicherung im Entwurf von SCR-Systemen. Es ergibt sich eine extreme Beschleunigung in der zuvor manuell durchgeführten Erstellung der CFD-Modelle. Im industriellen Einsatz sind die Kosten für Berechnungszeit und Infrastruktur wesentlich geringer zu veranschlagen als die Personalaufwendungen für die Erstellung der Modelle²⁰.

Auch wenn die volle Beschleunigung und Effizienzsteigerung der Entwurfssprachen erst beim Einsatz im Rahmen eines vollautomatisierten Bottom-Up-Entwurfes mit eingeschlossener Optimierung zum Tragen kommt, führt die dargestellte Entwurfssprache zu erheblichen Effizienzsteigerungen: Die Dauer der manuellen Erstellung und die Überprüfung des CFD-Modelles inklusive der Aufbereitung des CAD-Modelles reicht in der industriellen Praxis von Tagen bis in den Bereich einer Woche. Durch den Einsatz der Entwurfssprache kann man auf eine aufwändige manuelle Überprüfung der Einstellungen des CFD-Modelles verzichten. Darüber hinaus entfällt auch die zeitaufwändige Aufbereitung des CAD-Modelles durch die automatisierte Erstellung eines wasserdichten CAD-Modelles.

Damit reduzieren sich die benötigten Mannstunden durch den Einsatz von graphenbasierten Entwurfssprachen im SCR-Systementwurf auf einen Bruchteil der in der konventionellen Entwurfsmethodik benötigten Personalressourcen: Der Personaleinsatz von mehreren Tagen in der Modellerstellung und Überprüfung kann im Idealfall auf einige Stunden autonomer Computerlaufzeit der Entwurfssprache, mit nur sporadischer manueller Überwachung, reduziert werden. Zusammen mit den oben umrissenen Kostenunterschieden zwischen Personalstunde und Rechenstunde sind durch den Einsatz von Entwurfssprachen sehr große Produktivitätsgewinne zu erzielen, auch wenn die sehr zeitintensive manuelle Erstellung der Entwurfssprachen ebenfalls mitberücksichtigt werden.

²⁰Kosten für Rechenzeit sind aktuell (Stand 28.01.2014) grob im Bereich von 0.01\$ bis 1.00\$ pro Prozessorkern und Stunde anzusetzen (Quelle: PiCloud inc., www.picloud.com). Die Kosten für die Mannstunde eines Berechnungsingenieurs belaufen sich typischerweise auf 50 € bis 150 €.

Die zukünftig zu erwartenden weiter steigenden Rechenleistungen führen, speziell für die CFD-Berechnungen sowie für die automatisierte Verrohrung im Zusammenspiel mit einer weiteren Optimierung der verwendeten Algorithmen, mit großer Sicherheit auf ein Entwurfsszenario, in dem die Modellerstellung und Qualitätssicherung der virtuellen Modelle zum Flaschenhals des Entwurfsprozesses wird. An dieser Stelle können dann wissensbasierte Entwurfsmethoden wie die vorliegende Entwurfssprache, ihren Nutzen, über den hier demonstrierten industriellen Einsatz als effizienzsteigerndes Assistenzsystem, voll entfalten.

4. Verrohrungssprache

Für die computergestützte Generierung von Verrohrungen, bestehend aus einem Satz vorgegebener Standardrohrbauteile, wird in dieser Arbeit die Implementierung einer Verrohrungsentwurfssprache²¹ vorgeschlagen. Entsprechend Kapitel 2.4.2 finden sich in der Literatur verschiedene Ansätze zur automatisierten Generierung von Verrohrungsstrecken. Diese erlauben, nach dem aktuellen Wissensstand des Autors, nicht die Generierung beliebiger Rohrstrecken, welche gleichzeitig aus vorgegebenen Standardbauteilen bestehen. Bekannte Ansätze sind in der Regel mit Einschränkungen bezüglich der räumlichen Lage, beziehungsweise des räumlichen Verlaufes der Rohrstrecken, wie zum Beispiel in [7], verbunden.

4.1. Lösungsweg und Methodik

In den folgenden Abschnitten wird zunächst die Struktur im Überblick dargestellt. Anschließend wird die Repräsentation des Modelles der Verrohrungsstrecke und die einzelnen Regeln im Detail dargestellt. Abschließend wird die Regelabfolge der Verrohrungssprache in Form des entsprechenden Produktionssystems und die Integration des verwendeten Optimierungsalgorithmus dargestellt.

4.1.1. Rohrklassen und Regeln

Die Verrohrungssprache soll entsprechend dem in Kapitel 2.3 dargestellten Entwurfssprachenparadigma aufgebaut sein: Dazu wird ein Regelwerk analog zu den Gestaltsgrammatiken aus [187] definiert. Dieses Regelwerk soll aus Regeln zum Auf- und Umbau des Modells einer Rohrstrecke aus vorgegebenen Rohrbauteilen bestehen. Die Standardrohrbauteile stellen damit, neben den Umgebungsinformationen des Konfigurationsraumes (Bauraum, Neben-/Zwangsbedingungen bezüglich der Positionierung der Rohrstrecke), das Vokabular der Verrohrungssprache beziehungsweise in der UML-Notation das Klassendiagramm dar.

Abbildung 4.1 zeigt ein vereinfachtes Klassendiagramm der Verrohrungssprache mit den Start- und Endpositionen P_S und P_E sowie Richtungen \vec{D}_S und \vec{D}_E der zu bestimmenden Rohrstrecke in den Klassen *Start* und *Ziel*. Es enthält außerdem Klassen, welche Informationen über den Bauraum und sonstige Nebenbedingungen repräsentieren, sowie die zur Verfügung stehenden Standardrohrbauteile enthalten (im Klassendiagramm in Abbildung 4.1 Bögen mit Winkeln $\bar{\alpha}$ von 15° , 30° , 45° , 60° und 90°). Aus diesen Rohrbögen mit vorgegebenen Winkeln $\bar{\alpha}$ und gegebenen Biegeradien sowie aus geraden Rohrstücken mit variablen Längen soll mit dem Regelwerk der Verrohrungssprache eine optimale Verrohrung synthetisiert werden.

²¹Kurz: Verrohrungssprache

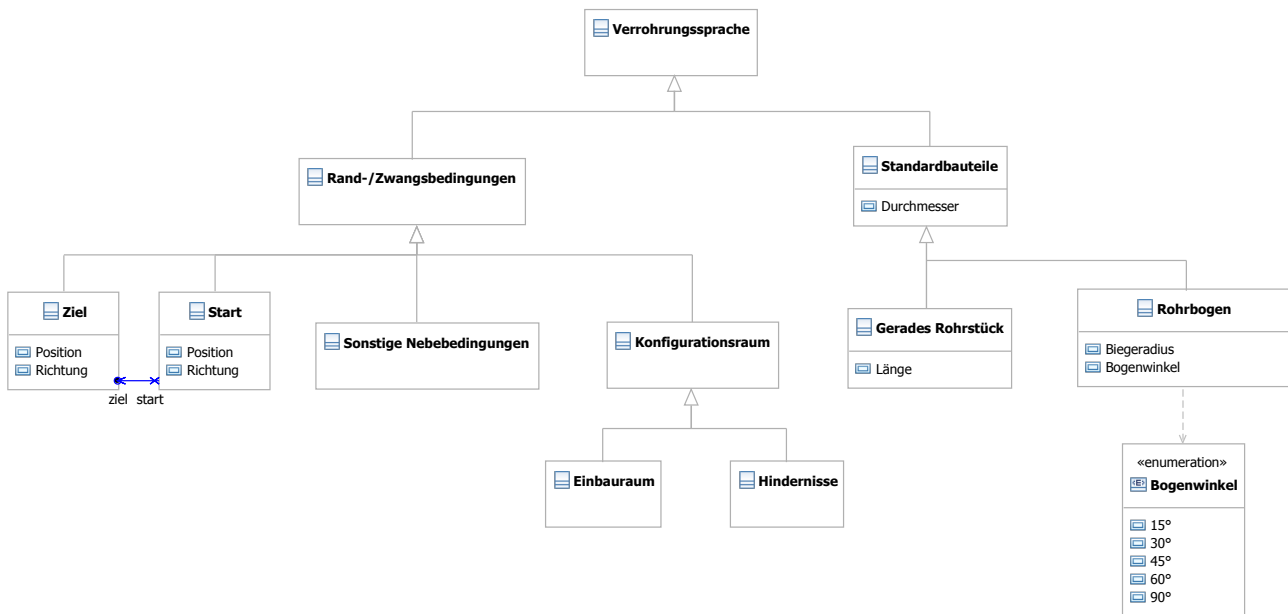


Abbildung 4.1: Vereinfachtes Klassendiagramm der Verrohrungssprache.

Die Rohrstrecke soll in der Verrohrungssprache als Polygonzug (Punktfolge) gespeichert werden. Folgende Regeln werden zum Auf- und Umbau der Rohrstrecke (=Polygonzug) definiert:

- Axiom: Initialisierung einer Verrohrung zwischen Start und Endpunkt unter Einhaltung der Randbedingungen von Start-/Endposition und Richtung sowie Einhaltung der zulässigen Bogenwinkel (*A* in Abbildung 4.2).
- Bogen einfügen: Hinzufügen eines Rohrbogens in der Rohrstrecke unter Berücksichtigung der zulässigen Bogenwinkel und Aufrechterhaltung der Start-/Endposition und Richtung (*B* in Abbildung 4.2).
- Winkel eines Bogens ändern: Änderung des Winkels eines Rohrbogens der Rohrstrecke unter Berücksichtigung der zulässigen Bogenwinkel und Aufrechterhaltung der Start-/Endposition und Richtung (*C* in Abbildung 4.2).
- Bogen löschen: Entfernen eines Bogens unter Beibehaltung der zulässigen Rohrbogenwinkel und Aufrechterhaltung der Start-/Endposition und Richtung (*D* in Abbildung 4.2).
- Verschiebung des Pfades: Verschiebung der Zwischenpunkte der Rohrstrecke unter Beibehaltung der zulässigen Rohrbogenwinkel und Aufrechterhaltung der Start-/Endposition und Richtung (*E* in Abbildung 4.2).

Die Gültigkeit der Randbedingungen bezüglich der Start-/Endpositionen und Richtungen sowie der Rohrbogenwinkel werden mit den dargestellten Regeln gewährleistet. In diesem Falle ist dann ein in der Verrohrungssprache gebildeter Satz, im Sinne des Entwurfssprachenparadigma, syntaktisch korrekt.

Abbildung 4.2 zeigt die beschriebenen Regeln schematisch. Die Regeln *A*, *B*, *D* ändern dabei die Topologie der Verrohrung. Regeln *C* und *E* entsprechen einer parametrischen Änderung. Die Regeln sind jeweils im Konfigurationsraum durch den die Verrohrungsstrecke repräsentierenden Polygonzug in einem Linke-Hand-Seite (LHS) / Rechte-Hand-Seite (RHS) Schema dargestellt. Die Bogenwinkel $\bar{\alpha}_i$ sollen den Winkeln der zur Verfügung stehenden Rohrbögen entsprechen.

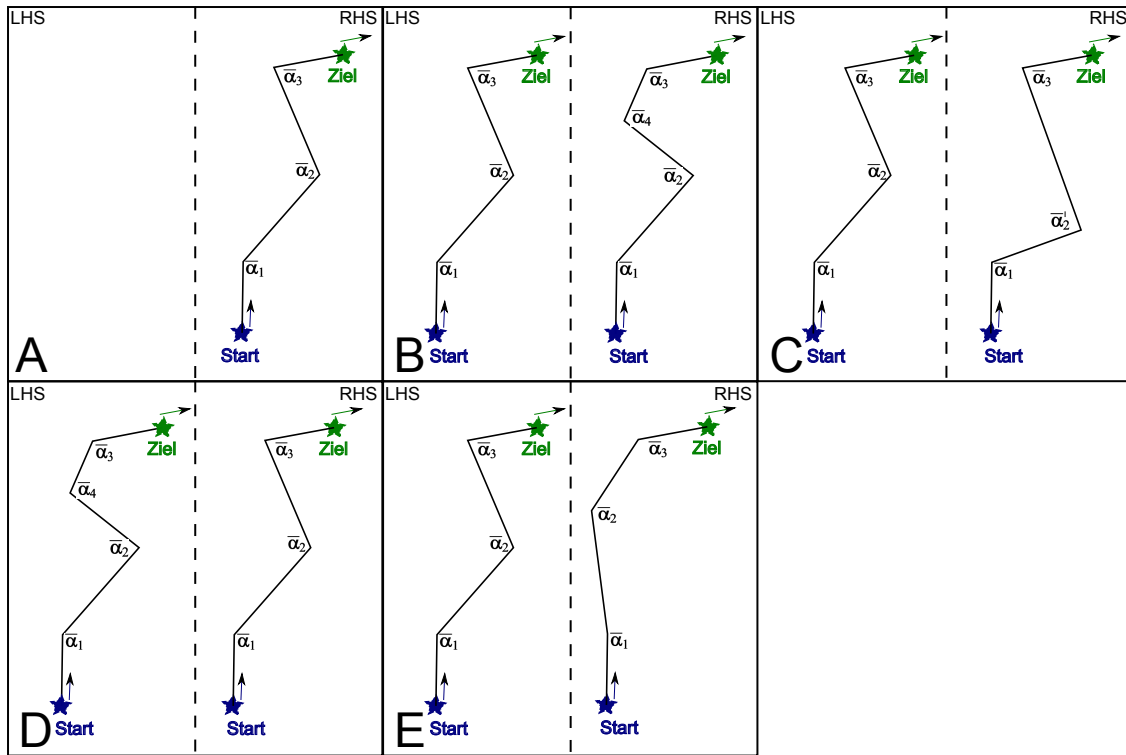


Abbildung 4.2: Schematische (Projektion in Zeichenebene!) Regeln der Verrohrungssprache im Konfigurationsraum aufgetragen (zugelassene Bogenwinkel $\bar{\alpha}_i$).

Regel *A* stellt als sogenanntes Axiom die erste auszuführende Regel der Verrohrungssprache dar, deren linke Seite (Prämisse) leer ist. Im Axiom wird der erste gültige Satz der Entwurfssprache im Entwurfsgraphen erzeugt. Die Regel muss damit ein erstes, mit den vorgegebenen Randbedingungen Start-/Endpositionen und Richtungen ($\vec{P}_S, \vec{P}_E, \vec{D}_S, \vec{D}_E$) sowie den vorgegebenen Rohrbogenwinkeln $\bar{\alpha}_i$ verträgliches, syntaktisch korrektes Modell der Rohrstrecke erzeugen. Diese Regel führt damit formal auf die Teilmenge der mit den gegebenen Randbedingungen vereinbarten Rohrstrecken des Entwurfsraumes. Alle weiteren Regeln *B-E* dienen dann der Manipulation der Rohrstrecke innerhalb dieser Teilmenge, unter Einhaltung der gegebenen Randbedingungen, und stellen damit dessen syntaktische Gültigkeit sicher.

Im Allgemeinen wird vom Entwerfer ein möglichst einfacher Entwurf angestrebt ("Keep it simple, stupid.", KISS-Prinzip [148]), der aus möglichst wenigen Rohrelementen bestehen soll. Zur Minimierung der Komplexität des ersten Entwurfes soll darum im Axiom möglichst eine Rohrstrecke mit minimaler Rohrbogenzahl und Winkelsumme initialisiert werden. Als untere Grenze der Winkelsumme dieser minimalen Konfiguration kann der Winkel α_{Min} zwischen Start- und Endrichtung als notwendige Bedingung abgeleitet werden:

$$\sum_{i|i \in Axiom} \bar{\alpha}_i \geq \alpha_{Min} = \angle(\vec{D}_S, \vec{D}_E). \tag{4.1}$$

Andere Nebenbedingungen und Optimierungsziele (Bauraumeinhaltung, minimale Rohrstreckenlänge, minimale Biegungswinkel,...), welche als semantische und pragmatische Aspekte der Verrohrungssprache bezeichnet werden (siehe auch Abbildung 6.2), sollen dann durch die nachfolgenden regelbasierten Manipulationen (Regeln *B-E*) der Rohrstrecke erfüllt werden.

4.1.2. Modellrepräsentation der Vokabeln im Starrkörpersimulator

Für die Implementierung der Verrohrungssprache wird ein Starrkörpersimulationsprogramm gewählt. Dieses bietet die Möglichkeit, die Bewegung starrer Körper, die über verschiedene Gelenke und Verbindungen gekoppelt sind, unter Berücksichtigung der sich aus den Kopplungen ergebenden Zwangsbedingungen, zu berechnen [205]. Programme zur Berechnung solcher Starrkörperbewegungen verwenden in der Regel numerische Differentialgleichungslöser zur simultanen Lösung der Bewegungsgleichungen und der Zwangsbedingungsgleichungen auf Basis des Lagrange-Formalismus sowie dem d'Alembertschen Prinzip der virtuellen Arbeit. Im Detail interessierten Lesern seien die Quellen [132, 205] empfohlen.

Das Modell der Rohrstrecke der Verrohrungssprache wird im Starrkörpersimulationsprogramm als über Gelenke verbundene Körper mit aufgeprägten Zwangskräften abgebildet. Die Verrohrung wird durch einen Polygonzug Π aus N Ortsvektoren $\vec{P}_i = \vec{P}_i(x_i, y_i, z_i)$ mit $i \in [1, 2, \dots, N]$ und den jeweiligen Richtungsvektoren $\vec{D}_j = \vec{P}_{i+1} - \vec{P}_i$ mit $j \in [1, 2, \dots, N - 1]$ repräsentiert. Die Winkel zwischen benachbarten Richtungsvektoren lassen sich durch $\alpha_k = \angle(\vec{D}_k, \vec{D}_{k+1})$ mit $k \in [1, 2, \dots, N - 2]$ berechnen (siehe auch Abbildung 4.3). Die Indizes k der Winkel stellen gleichzeitig eine Nummerierung der Rohrbögen dar. Daraus lassen sich folgende Zwangsbedingungen in der Modellierung der Rohrstrecke formulieren (das hochgestellte $^\circ$ symbolisiert einen normierten Vektor: $\vec{v}^\circ = \frac{\vec{v}}{\sqrt{\vec{v} \cdot \vec{v}}}$):

$$\alpha_k - \bar{\alpha}_j = 0, \quad (4.2)$$

$$\vec{P}_S - \vec{P}_0 = \vec{0}, \quad (4.3)$$

$$\vec{P}_E - \vec{P}_N = \vec{0}, \quad (4.4)$$

$$\vec{D}_S^\circ - \vec{D}_0^\circ = \vec{0}, \quad (4.5)$$

$$\vec{D}_E^\circ - \vec{D}_{N-1}^\circ = \vec{0}. \quad (4.6)$$

Als letzte Zwangsbedingung muss berücksichtigt werden, dass die Rohrbögen einer realen Verrohrung eine Ausdehnung besitzen, die durch die Eckpunkte \vec{P}_1 bis \vec{P}_{N-1} des Polygonzuges noch nicht repräsentiert ist. Abbildung 4.3 stellt den Platzbedarf eines Rohrbogens dar.

Aus der Konstruktion in Abbildung 4.3 lässt sich die im Polygonzug für die Ausdehnung der Rohrbögen zu reservierende Strecke mit Gleichung (4.7) berechnen:

$$X_i = \tan\left(\frac{\bar{\alpha}_i}{2}\right) \cdot R_i. \quad (4.7)$$

Daraus ergeben sich folgende zusätzliche Zwangsbedingungen:

$$\sqrt{(\vec{P}_1 - \vec{P}_2)^2} \geq \tan\left(\frac{\bar{\alpha}_1}{2}\right) \cdot R_1 \quad \text{für } i = 1 \quad (4.8)$$

$$\sqrt{(\vec{P}_{i+1} - \vec{P}_i)^2} \geq \tan\left(\frac{\bar{\alpha}_{i-1}}{2}\right) \cdot R_{i-1} + \tan\left(\frac{\bar{\alpha}_i}{2}\right) \cdot R_i \quad \text{für } 1 < i < N - 1 \quad (4.9)$$

$$\sqrt{(\vec{P}_{N-1} - \vec{P}_N)^2} \geq \tan\left(\frac{\bar{\alpha}_{N-2}}{2}\right) \cdot R_{N-2} \quad \text{für } i = N - 1. \quad (4.10)$$

In der vorliegenden Arbeit wird für die Implementierung die Java-Portierung ODE4J [210] des Starrkörpersimulationsprogrammes ODE [177] verwendet. Diese Softwarebibliothek enthält neben der Starrkörpersimulation auch Funktionen zur Kollisionsdetektion zwischen den simulierten Starrkörpern. Die Kollisionsdetektion findet parallel zur Lösung und Evolution der

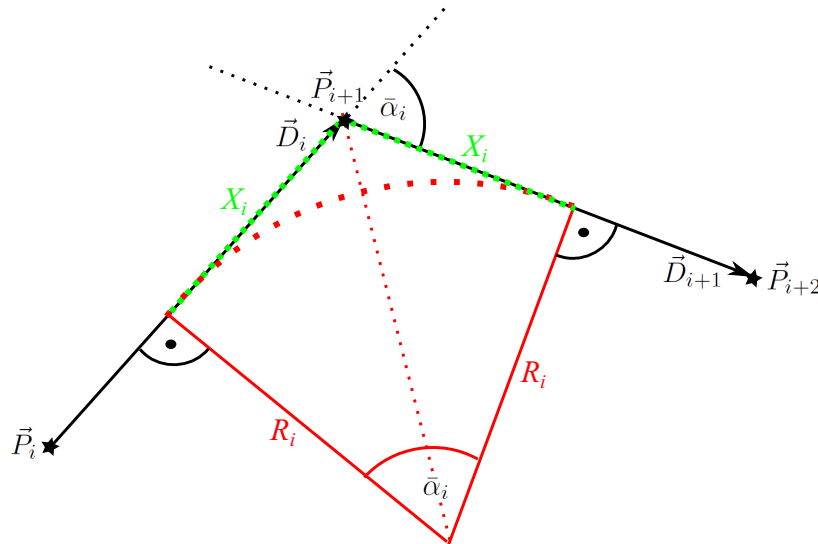


Abbildung 4.3: Darstellung des Platzbedarfes X_i (grün gestrichelt) des Rohrbogens i mit Biegeradius R_i und Bogenwinkel $\bar{\alpha}_i$. Polygonzug ist in schwarz dargestellt. Die Mittellinie des Rohrbogens ist dick rot gepunktet angedeutet.

Bewegungsgleichungen statt [205]. Kommt es zum Kontakt zwischen den starren Körpern, so wird die Berechnung der Bewegungsgleichungen angehalten und der neue Zustand der Körper nach der Kollision berechnet. Mit diesen neuen Zuständen wird die Berechnung der Bewegungsgleichungen anschließend wieder aufgenommen. Für Körper, die in Kontakt bleiben, werden entsprechende Zwangskräfte in die Bewegungsgleichungen eingeführt [205]. Dieser Zugang erlaubt später eine simultane beziehungsweise Behandlung von mehreren Rohrlinien im gleichen "Universum" des Starrkörpersimulators.

Die Kollisionsdetektion und Kollisionsbehandlung des Starrkörpersimulators kann für die Einhaltung der $N - 1$ Zwangsbedingungen (4.8) bis (4.10) verwendet werden. Dazu werden die Eckpunkte des Polygonzuges \vec{P}_2 bis \vec{P}_{N-1} und damit die Rohrbögen durch starre Kugeln approximiert, welche sich gegenseitig nicht durchdringen können. Sie sollen an der Position \vec{P}_{i+1} einen Radius von X_i entsprechend Gleichung (4.7) besitzen. Dies stellt den Mindestabstand und damit den Platzbedarf der Rohrbögen entsprechend den Zwangsbedingungen (4.8) bis (4.10) sicher. In erster Näherung sollen Kugeln verwendet werden, da diese bereits implementiert sind und durch die analytische Formulierung der Geometrie in der Kollisionsdetektion besonders effizient behandelt werden können.

Die geraden Rohrstrecken zwischen den Rohrbögen, beziehungsweise zwischen den Kugeln im Starrkörpersimulator, werden durch Zylinder, die im Starrkörpersimulator zur Verhinderung einer Durchdringung ebenfalls gegenseitig kollidieren können, repräsentiert. Bei vorgegebenen Bogenwinkeln $\bar{\alpha}_i$ bleiben damit noch folgende Freiheitsgrade in der Rohrstrecke übrig:

1. Längenänderung der geraden Rohrstrecken. Untere Grenze der Länge der Polygonzugstrecken durch den Radius der Platzhalter-Kugeln im Zusammenspiel mit der Kollisionsdetektion entsprechend den Zwangsbedingungen (4.8) bis (4.10) sicher gestellt.
2. Rotation der Rohrbögen um die Achsen der jeweils angrenzenden geraden Rohrstücke beziehungsweise Polygonzugstrecken.

Diese Freiheitsgrade werden im Starrkörpersimulator durch die Modellierung der geraden Rohrstücke zwischen den Eckpunkten beziehungsweise Kugeln durch Dreh-Schiebe-Gelenke (Piston

Joint, beziehungsweise Zylindergelenk) realisiert. Dreh-Schiebe-Gelenke erlauben eine relative Rotation zwischen den gekoppelten, coaxialen Achsen sowie eine axiale Verschiebung entlang der gekoppelten Achsen (Abbildung 4.4).

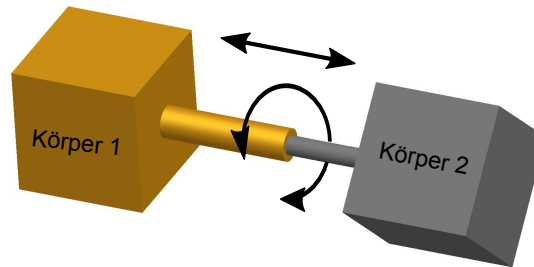


Abbildung 4.4: Freiheitsgrade eines Dreh-Schiebe-Gelenkes.

Die Enden dieser Dreh-Schiebe-Gelenke sind starr mit den Eckpunkten \vec{P}_{i+1} des Polygonzuges, beziehungsweise den Kugeln verbunden und gewährleisten so die Aufrechterhaltung des (Bogen-)Winkels $\bar{\alpha}_i$ zwischen den ein- und ausgehenden geraden Rohrstücken des jeweiligen Rohrbogens. Abbildung 4.5 stellt schematisch die Darstellung der Rohrstrecke als Polygonzug und die Repräsentation der Rohrstrecke im Starrkörpersimulationsprogramm gegenüber.

4.1.3. Umsetzung der Regeln im Starrkörpersimulator

In diesem Abschnitt wird die Umsetzung und Implementierung der zuvor skizzierten Regeln auf Basis der im letzten Abschnitt dargestellten Modellrepräsentation dargestellt.

Regel A: Axiom Im Axiom muss gewährleistet werden, dass eine Verrohrung initialisiert wird, welche mit den Zwangsbedingungen (4.2) bis (4.10) verträglich ist. Unter Berücksichtigung der unteren Grenze der Bogenwinkelsumme aus Gleichung (4.1) soll eine Kombination aus Rohrbögen vorgegeben werden können, die im Axiom zu einer mit den Zwangsbedingungen kompatiblen Verrohrung zusammengesetzt wird.

Die initiale Konfiguration soll als geordnete Liste der zu initialisierenden Rohrbögen $I = \{\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_{N-2}\}$, beginnend mit der Nummerierung an Startposition \vec{P}_S , mit dem Winkel des ersten Rohrbogens $\bar{\alpha}_1$, vorgegeben werden. Eine entsprechende Liste R wird für die Biegeradien R_i der jeweiligen Rohrbögen definiert und vorgegeben. Beide Listen sollen zusammen die Rohrbogenkonfiguration (I, R) der Rohrstrecke bilden.

Nicht für alle Zwangsbedingungen (4.2)-(4.10) und Rohrbogenkonfigurationen (I, R) existiert eine valide Rohrstrecke. Dies soll anhand eines kleinen Beispiels in Abbildung 4.6 veranschaulicht werden. Gezeigt ist die Start- und Endposition \vec{P}_S und \vec{P}_E der Verrohrung sowie die Start- und Endrichtungen \vec{D}_S und \vec{D}_E der Verrohrung mit einer initialen Rohrbogenkonfiguration (I, R) mit $|I| = |R| = 2$ und damit $N = 4$. Auf Grund des geringen Abstandes von Start- und Endposition existiert kein ausreichender Platz für das Unterbringen der Rohrbögen. Dieses Beispiel stellt nur eine mögliche Situation dar, in der auf Grund der Unverträglichkeit der gegebenen Zwangsbedingungen keine Rohrkonfiguration gefunden werden kann.

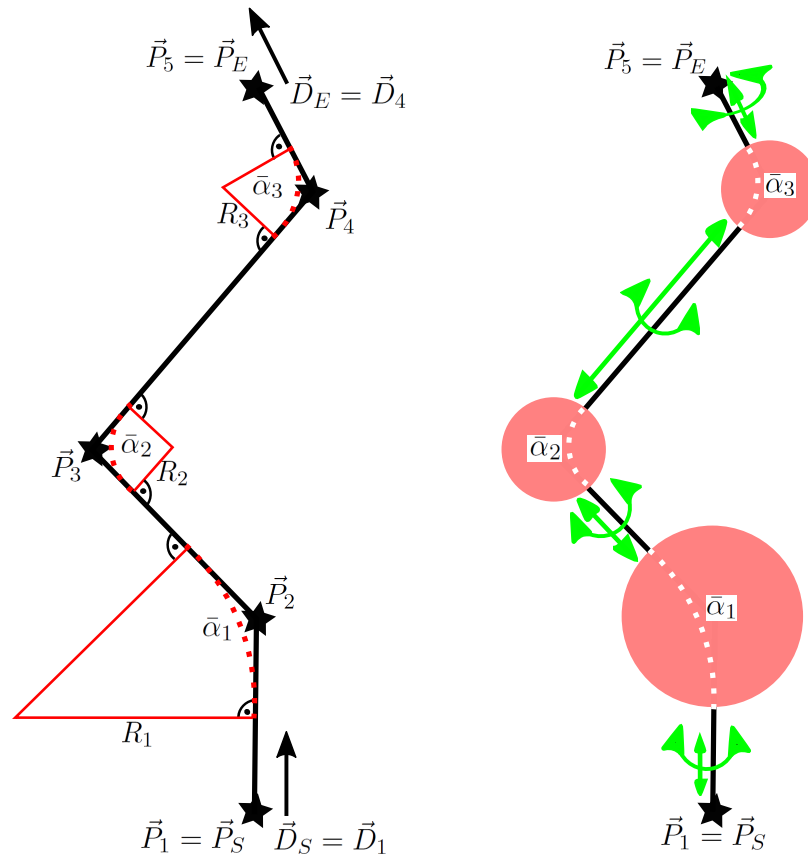


Abbildung 4.5: Schematische Gegenüberstellung der Rohrstrecke als Polygonzug (links) und die Repräsentation der Rohrstrecke im Starrkörpersimulationsprogramm (rechts). Grün markiert sind die Freiheitsgrade der Verrohrung, welche im Starrkörpersimulator als Dreh-Schiebe-Gelenke abgebildet sind. Rot dargestellt sind die Platzhalter-Kugeln an den Positionen der Rohrbögen.

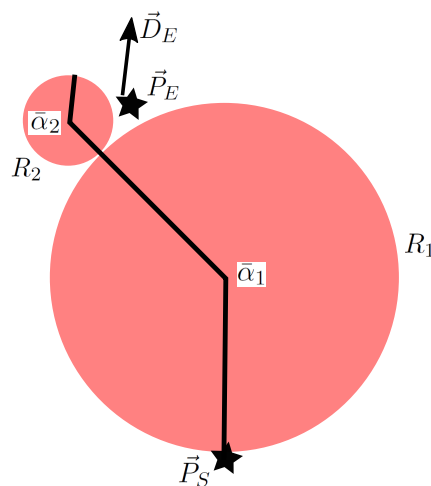


Abbildung 4.6: Beispiel unverträglicher Zwangsbedingungen für die keine Rohrstrecke gefunden werden kann: Endrichtung \vec{D}_E ist erfüllt, aber Endposition \vec{P}_E kann auf Grund der zu berücksichtigten Ausdehnung der Rohrbögen nicht erreicht werden.

Dies bedeutet, dass im Axiom bestimmt werden sollte, ob unter den gegebenen Randbedingungen überhaupt eine Rohrstrecke existiert. Dazu werden im Starrkörpersimulator folgende Schritte ausgeführt:

1. Initialisierung einer (zufälligen) Rohrstrecke mit der Rohrbogenkonfiguration (I, R) unter Einhaltung der Zwangsbedingungen Startposition (4.3) und Startrichtung (4.5). Dies stellt eine einseitig an der Startposition fixierte Rohrstrecke mit einem freien Ende dar.
2. Bringe Kraft \vec{F}_{posE} auf das freie Ende der Rohrstrecke \vec{P}_N in Richtung des Endpunktes auf $(\vec{F}_{posE} || \vec{P}_E - \vec{P}_N)$ bis Zwangsbedingung (4.4), Einhaltung der Endposition, erfüllt ist.
3. Fixierung der Endposition $\vec{P}_N = \vec{P}_E$ mit einer Kugelgelenk-Lagerung: Keine Verschiebungsfreiheitsgrade, volle Rotationsfreiheitsgrade.
4. Bringe Kraft \vec{F}_{ricE} auf den letzten Rohrbogen \vec{P}_{N-1} in Richtung der Achse \vec{D}_E auf, der Kraftvektor \vec{F}_{ricE} sei parallel zur Projektion von $\vec{D}_{N-1} - \vec{D}_E$ auf die Normalenebene von \vec{D}_E , bis Zwangsbedingung (4.6), die Einhaltung der Endrichtung, erfüllt ist.
5. Ersetzung der Kugelgelenk-Lagerung der Endposition, nun mit $\vec{P}_N = \vec{P}_E$ und $\vec{D}_{N-1} = \vec{D}_E$, durch eine fixierte Lagerung: Keine Verschiebungs- und Rotationsfreiheitsgrade.

Die Abfolge dieser Einzelschritte stellt eine Heuristik zum Auffinden einer ersten, mit den Zwangsbedingungen kompatiblen, Rohrstrecke dar. Mit der Initialisierung einer zufälligen Rohrstrecke mit einem freien Ende kann es zu Konstellationen kommen, welche, trotz theoretisch möglicher Existenz einer mit den Zwangsbedingungen (4.4) und (4.6) verträglichen Rohrstrecke, das Auffinden dieser Lösung mit dem beschriebenen Regelwerk verhindern. Zur Veranschaulichung ist beispielhaft ein solcher degenerierter Fall in Abbildung 4.7 dargestellt und in der Bildunterschrift erklärt. Solche degenerierten Fälle sind ein in der Berechnung von kinematischen Ketten auftretender Effekt: Sobald zuvor linear unabhängige Freiheitsgrade, durch eine ungünstige Stellung der Gelenke, linear abhängig voneinander werden (z.B. Parallelstellung von Gelenkachsen) kommt es zum Auftreten von Singularitäten in den Bewegungsgleichungen [129].

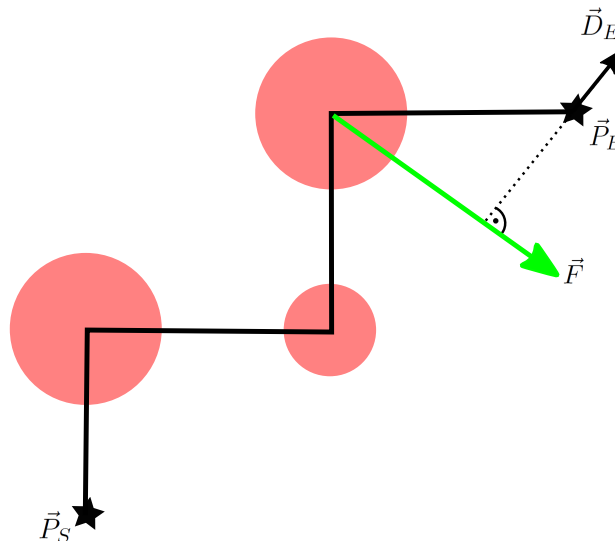


Abbildung 4.7: Verlust von Freiheitsgraden durch die Lage des Kraftvektors $\vec{F} = \vec{F}_{ricE}$ in einer Ebene mit den benachbarten Rohrelementen. Dies führt zu einer Blockade des Systems innerhalb der Ebene, obwohl eine Lösung, außerhalb der Ebene, existieren würde.

Alternativ könnte im Axiom eine Initialisierung einer mit den Zwangsbedingungen verträglichen Rohrstrecke über eine direkte Berechnung der Rohrstreckenparameter analog zur inversen Kinematikberechnung bei Robotern erfolgen [129, 36]. Dies ist aber wesentlich zeitaufwändiger als eine zufällige Initialisierung und durch das mögliche Auftreten von Singularitäten, analog zu dem oben beschriebenen degenerierten Fall, ebenfalls nicht frei von Unwägbarkeiten [129].

Im Laufe der Erstellung dieser Arbeit wurde zunächst ein alternativer Ansatz der Initialisierung einer ersten Rohrstrecke, die mit den Randbedingungen kompatibel ist, versucht. Dazu wurde ein genetischer Algorithmus [208] eingesetzt. Darin wurden die Zwischenpunkte des Polygonzuges im Fortlauf der Optimierung variiert. Die Nutzenfunktion wurde so gewählt, dass die Abweichung der Start- und Endrichtung des Polygonzuges sowie die Abweichungen der Winkel von den gegebenen Größen minimiert werden sollten.

Um eine mit den Randbedingungen kompatible Konfiguration zu finden, müssen mit diesem Ansatz die Abweichungen der Richtungen und Winkel zu den gegebenen Größen unter einen Toleranzwert gebracht werden. Dieser Toleranzwert wird sinnvollerweise im Bereich der geometrischen Toleranz des CAD-Kernels [136] gewählt. Der evolutionäre Optimierungsvorgang hatte viel längere Laufzeiten als die oben genannte Initialisierungsheuristik im Starrkörpersimulator.

Dies könnte daran liegen, dass im Starrkörpersimulator die kinematischen Freiheitsgrade implizit abgebildet sind, die Randbedingungen der Rohrbogenwinkeln schon von Anfang an erfüllt sind und in der Initialisierung 'nur' noch die Zielposition und Richtung eingestellt werden müssen. Dagegen ist der Suchraum bei Verwendung des genetischen Algorithmus wesentlich größer: Die Randbedingung der Endposition ist zwar erfüllt, die Start- und End-Richtung sowie die Rohrbogenwinkel müssen in der Optimierung aber erst gefunden werden. Gleiches gilt für die Berücksichtigung der Ausdehnung der Rohrbögen, welche bei der Initialisierung im Starrkörpersimulator explizit in der Kinematik, über die Rückwirkung der Kollisionsdetektion auf die Bewegung der Rohrstrecke, Eingang findet.

Regel B: Bogen hinzufügen (KKF-Konstruktion) In dieser Regel soll ein zusätzlicher Bogen in die Rohrstrecke, anstelle eines bestehenden geraden Streckenelements, eingefügt werden. Die Regel *B* soll auf eine eingefrorene Rohrstrecke, die sich im Starrkörpersimulator in Ruhe befindet, angewandt werden. Vor der Regelausführung wird die Simulation dazu im Starrkörpersimulator gestoppt und erst nach erfolgter Regelanwendung wieder aufgenommen.

Wie oben dargestellt (Abbildung 4.6) kann es zu Situationen kommen, in denen die gegebenen Zwangsbedingungen im gegenseitigen Widerspruch stehen und so keine gültige Rohrstreckenkonfiguration gefunden werden kann. Dasselbe gilt auch für das Einfügen von Rohrbögen in die eingefrorene Rohrstrecke. Ist beispielsweise in der bestehenden Rohrstrecke kein ausreichender Abstand zwischen benachbarter Rohrbögen vorhanden, um den einzufügenden Rohrbogen aufzunehmen, so kann die Regel auf die gegebene Rohrstreckenkonfiguration nicht angewandt werden. In der folgenden Darstellung werden weitere geometrische Situationen betrachtet, die dem Einfügen eines Rohrbogens an einer gegebenen Position der Rohrstrecke entgegenstehen.

Im ersten Schritt soll gezeigt werden, wie die mögliche(n) Einbauposition(en) eines neuen Rohrbogens, unter Einhaltung der Zwangsbedingungen, geometrisch bestimmt werden können. Ziel soll es sein, einen neuen Rohrbogen zwischen den bestehenden Rohrbögen $i_{\leftarrow} = k$ und $i_{\rightarrow} = k+1$ einzufügen. Die Rohrstrecke soll nach erfolgter Einfügung mit dem Index i' neu nummeriert werden. Das Hochkomma kennzeichnet so den Zustand nach erfolgreicher Regelausführung. Somit ergibt sich für die zu betrachtende Rohrstecke folgende neue Indizierung:

- Vorheriger Rohrbogen: $i'_{\leftarrow} = k$
- Neu eingefügter Rohrbogen: $i'_1 = k + 1$
- Folgender Rohrbogen: $i'_{\rightarrow} = k + 2$

Der Index aller nachfolgender Rohrbögen i'_{\rightarrow} mit $i_{\rightarrow} > k + 1$ wird entsprechend um eins erhöht $i'_{\rightarrow} = i_{\rightarrow} + 1$. Entsprechend sollen auch die Orts- beziehungsweise Richtungsvektoren $\vec{P}_i \rightarrow \vec{P}_{i'}$ und $\vec{D}_i \rightarrow \vec{D}_{i'}$ sowie die Rohrbogenkonfiguration (I, R) neu indiziert werden.

Für die geometrische Bestimmung der Lage des neu einzufügenden Rohrbogens soll zunächst die Punktmenge der möglichen Positionen $\vec{P}_{i'_1}$ des einzufügenden Rohrbogens bestimmt werden. Diese Punktmenge soll die Einhaltung der zugelassenen Rohrbogenwinkel $\bar{\alpha}_{i_{\leftarrow}}$ beziehungsweise $\bar{\alpha}_{i_{\rightarrow}}$, der an die zu ersetzende gerade Rohrstrecke angrenzenden Rohrbögen, berücksichtigen. Diese Punktmenge stellt gerade zwei Kegel mit Kegelwinkel $2\bar{\alpha}_{i_{\leftarrow}=k}$ beziehungsweise $2\bar{\alpha}_{i_{\rightarrow}=k+1}$, Scheitelpunkt \vec{P}_{k+1} beziehungsweise \vec{P}_{k+2} und Achse \vec{D}_k beziehungsweise $-\vec{D}_{k+2}$ dar (Indexierung vor Einfügung). Abbildung 4.8 zeigt diese Situation schematisch.

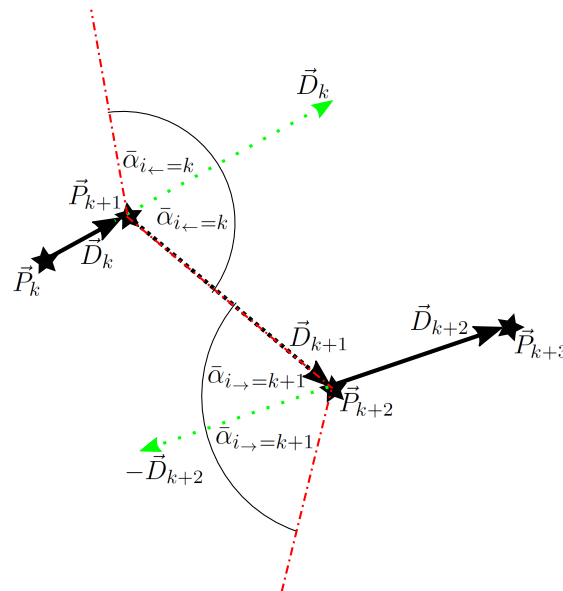


Abbildung 4.8: Die Abbildung zeigt die Punktmenge der möglichen Positionen des einzufügenden Rohrbogens, welche die vorgegebenen Rohrbogenwinkel $\bar{\alpha}_{i_{\leftarrow}}$ beziehungsweise $\bar{\alpha}_{i_{\rightarrow}}$ gewährleisten: Das schwarz gepunktete Streckenelement $\overline{P_{k+1}P_{k+2}}$ soll durch einen zusätzlichen Rohrbogen, mit zwei neuen Streckenelementen, ersetzt werden. Die Kegelmäntel mit Kegelwinkel $2\bar{\alpha}_{i_{\leftarrow}=k}$ beziehungsweise $2\bar{\alpha}_{i_{\rightarrow}=k+1}$, Scheitelpunkt \vec{P}_{k+1} beziehungsweise \vec{P}_{k+2} und Achse \vec{D}_k beziehungsweise $-\vec{D}_{k+2}$ (grün) sind rot gepunktet angedeutet.

Die Schnittmenge der beiden Kegelmäntel bildet dann die möglichen Positionen $\vec{P}_{i'_1}$ des neu einzufügenden Rohrbogens, welche die Rohrbogenwinkel $\bar{\alpha}_{i_{\leftarrow}}$ und $\bar{\alpha}_{i_{\rightarrow}}$ sicher stellen. Diese Schnittmenge muss im weiteren Verlauf noch nach der Teilmenge der Positionen, die mit der Ausdehnung der Rohrbögen (Nebenbedingungen (4.8)-(4.10)) verträglich sind, gefiltert werden.

Abbildung 4.8 zeigt, dass das zu ersetzende Streckenelement $\overline{P_{k+1}P_{k+2}}$ Teil dieser Schnittmenge ist, wenn man bei Anwendung der vorliegenden Regel B unveränderte Rohrbogenwinkel $\bar{\alpha}_{i_{\leftarrow}}$ und $\bar{\alpha}_{i_{\rightarrow}}$, der benachbarten Rohrbögen, fordert. An dieser Stelle soll nicht unerwähnt bleiben, dass diese beiden Winkel bei der Anwendung der betrachteten Regel, unter Beachtung der

zugelassenen Bogenwinkel $\bar{\alpha}$, in einer alternativen Regelformulierung auch gleichzeitig neu gewählt werden könnten. Dies würde das Einfügen eines neuen Rohrbogens und die gleichzeitige Modifikation eines, beziehungsweise beider Bogenwinkel der benachbarten Rohrbögen erlauben. In diesem Falle wäre das zu ersetzende Streckenelement nicht mehr Teil der Schnittmenge der möglichen Positionen des einzufügenden Rohrbogens, da die betrachteten Kegelmäntel entsprechend modifizierte Kegelwinkel besäßen.

Die bisher betrachtete geometrische Konstruktion berücksichtigt nur die Einhaltung der dem einzufügenden Rohrbogen benachbarten Bogenwinkel $\bar{\alpha}_{i\leftarrow}$ und $\bar{\alpha}_{i\rightarrow}$. Der einzufügende Rohrbogen soll aber ebenfalls einen fest vorgegebenen Rohrbogenwinkel $\bar{\alpha}_{i'}$ besitzen. Geometrisch formuliert bedeutet dies, dass die Position $\vec{P}_{i'+1}$ so bestimmt werden muss, dass die Strecken

$$\overline{P_{i'\leftarrow+1}P_{i'+1}} \quad (4.11)$$

$$\overline{P_{i'+1}P_{i'\rightarrow+1}} \quad (4.12)$$

am Scheitel $\vec{P}_{i'+1}$ gerade einen Winkel von

$$\angle(P_{i'\leftarrow+1}P_{i'+1}P_{i'\rightarrow+1}) = \bar{\alpha}_{i'} \quad (4.13)$$

einschließen.

In der Geometrie der Ebene ist eine entsprechende Konstruktion, zur Bestimmung der möglichen Positionen der Scheitelpunkte B bei vorgegebenen Winkel $\beta = \angle(ABC)$ sowie gegebenen Punkten A und C als Peripheriewinkelsatz oder Umfangswinkelsatz, [135, 108] beziehungsweise auch als Fasskreisbogenkonstruktion bekannt. Abbildung 4.9 zeigt die entsprechende Konstruktion. Für $\beta = 90^\circ$ ergibt sich der Satz von Thales. Der betrachtete Kreis ist Umkreis des Dreiecks ABC beziehungsweise $AB'C$. Die Strecke \overline{AC} bildet eine Kreissehne, die den Kreisbogen in zwei Segmente teilt. Der Punkt B beziehungsweise B' soll dabei auf den jeweiligen Kreissegmenten liegen. Der Peripherie- oder Umfangswinkel β beziehungsweise β' ist dann bei Verschiebung des Punktes B beziehungsweise B' auf dem jeweiligen Kreissegment konstant. Dabei gilt nach [108]:

$$\beta + \beta' = 180^\circ. \quad (4.14)$$

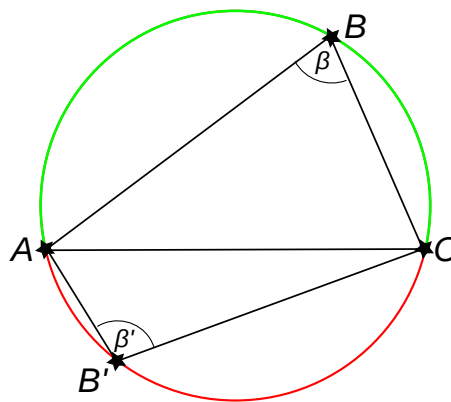


Abbildung 4.9: Darstellung des Peripheriewinkelsatzes: Bei Verschiebung von B beziehungsweise B' auf dem jeweiligen Kreissegment (grün beziehungsweise rot) bleibt β beziehungsweise β' konstant.

Im vorliegenden Fall der Verrohrungen ergibt sich der Winkel β der Fasskreisbogenkonstruktion gerade zu:

$$\beta = 180^\circ - \bar{\alpha}_{i'}. \tag{4.15}$$

Damit ergibt sich im ebenen Fall, aus dem Peripheriewinkelsatz und der Fasskreisbogenkonstruktion das in Abbildung 4.9 gezeigte Kreissegment als Menge der Positionen des Rohrbogens, welche den gewünschten Rohrbogenwinkel $\bar{\alpha}_{i'}$ garantieren.

Überträgt man diese Ergebnisse auf den räumlichen Fall, so führt eine Rotation des Kreissegmentes um die Sehne $\overline{AB} = \overline{P_{i'_{\leftarrow}+1}P_{i'_{\rightarrow}+1}}$ auf eine geschlossene Oberfläche, welche die gesuchte Menge aller Punkte $\vec{P}_{i'+1}$ darstellt, die den vorgegebenen Bogenwinkel $\bar{\alpha}_{i'}$ des einzufügenden Bogens (Bedingung (4.13)) sicherstellen. Abbildung 4.10 zeigt solche Fasskreisbögen für verschiedene Winkel $\bar{\alpha}_{i'}$ in der Ebene und im Raum rotiert dargestellt.

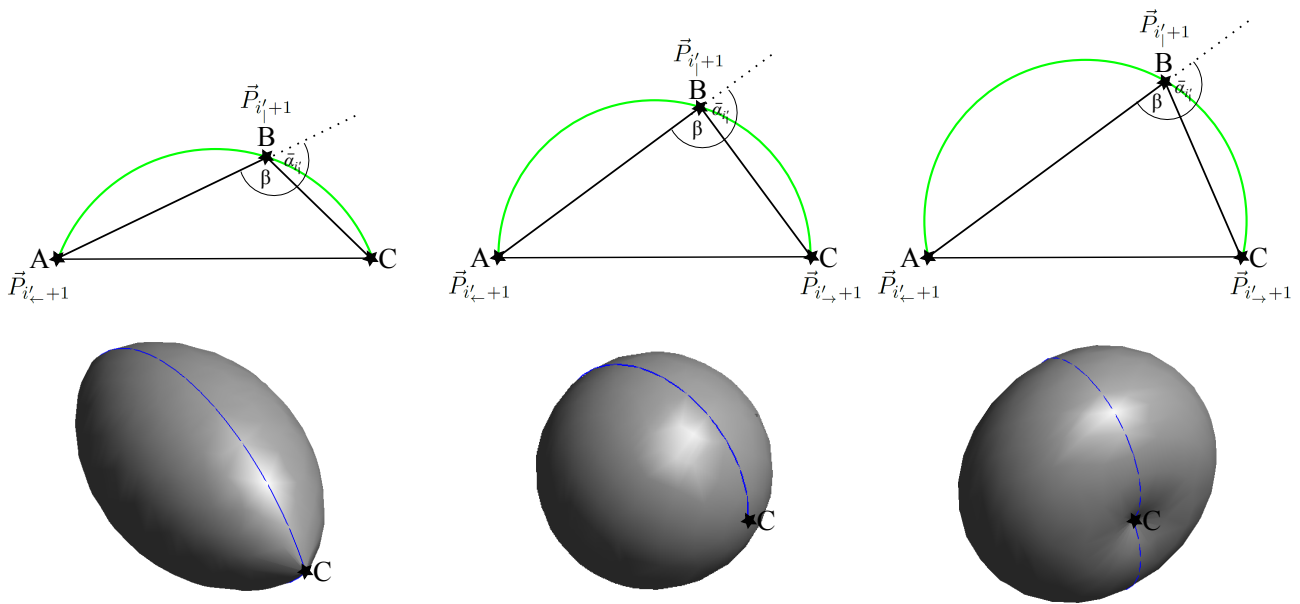


Abbildung 4.10: Fasskreisbögen beziehungsweise um \overline{AC} rotierte Fasskreisbögen mit $\bar{\alpha}_{i'} < 90^\circ$, $\bar{\alpha}_{i'} = 90^\circ$ und $\bar{\alpha}_{i'} > 90^\circ$ (von links nach rechts). In der Ebene (oben) beziehungsweise räumlich aufgetragen (unten).

Zur endgültigen Bestimmung der mit den vorgegebenen Bogenwinkeln kompatiblen Positionen muss die Schnittmenge der drei betrachteten Flächen (zwei Kegelmäntel und rotierter Fasskreisbogen) gebildet werden. Die Punkte dieser Schnittmenge, welche auf allen drei Flächen liegen, bilden mögliche Einbaupositionen. Die Ausdehnung der Rohrbögen, Zwangsbedingungen (4.8)-(4.10), wurde ja oben schon bei der Bestimmung der Kegelmäntel berücksichtigt. Abbildung 4.11 zeigt eine solche Konstruktion im Raum. Diese Konstruktion soll im Folgenden vereinfacht als *Kegel-Kegel-Fasskreisbogen (KKF)-Konstruktion* bezeichnet werden.

Bei Betrachtung von Abbildung 4.11 wird klar, dass es bei einer Änderung der beteiligten Parameter zu einer Verschiebung oder Deformation der drei zu verschneidenden Flächen kommen kann, so dass keine gemeinsame Schnittmenge der drei Flächen existiert. Dies stellt gerade der Fall sich ausschließender Zwangsbedingungen dar.

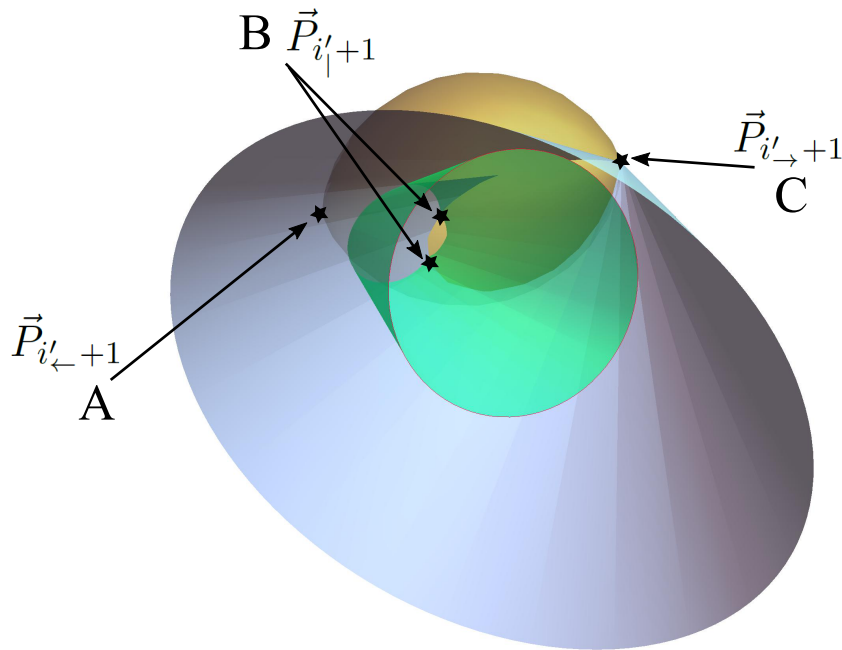


Abbildung 4.11: KKF-Konstruktion zur Bestimmung der möglichen Einbaupositionen $\vec{P}'_{i_{|}+1}$ (im Beispiel zwei Punkte) von Rohrbögen beim Hinzufügen eines Bogens in einer Rohrstrecke unter Berücksichtigung vorgegebener Rohrbogenwinkel.

Regel C: Winkel eines Rohrbogens ändern Wie in Regel B im Abschnitt 4.1.3 wird zum Zeitpunkt der Regelausführung im Starrkörpersimulator von einer eingefrorenen Rohrstrecke ausgegangen, deren Elemente sich im Stillstand befinden. Die Regel soll die Modifikation des Bogenwinkels $\bar{\alpha}_{i_{|}}$ des Rohrbogens $i = i_{|}$ erlauben. Der neue Bogenwinkel $\bar{\alpha}'_{i_{|}}$ sei wiederum aus der Liste der zugelassenen Rohrbogenwinkel $\bar{\alpha}$.

Es wird analog zur Regel B die KKF-Konstruktion mit den entsprechenden Rohrbogenwinkeln $\bar{\alpha}'_{i_{\leftarrow}} = \bar{\alpha}_{i_{\leftarrow}}$, $\bar{\alpha}'_{i_{|}}$ und $\bar{\alpha}'_{i_{\rightarrow}} = \bar{\alpha}_{i_{\rightarrow}}$ mit $i_{\leftarrow} = i_{|} - 1$ und $i_{\rightarrow} = i_{|} + 1$ angewandt. Dabei wird das Streckenstück $\overline{P_{i_{\leftarrow}+1}P_{i_{|}+1}P_{i_{\rightarrow}+1}}$ der Rohrstrecke unter Beibehaltung der Positionen $\vec{P}_{i_{\leftarrow}+1}$ und $\vec{P}_{i_{\rightarrow}+1}$ neu bestimmt.

Mit der KKF-Konstruktion werden die mit den Rohrwinkelzwangsbedingungen verträglichen neuen Positionen $\vec{P}'_{i_{|}+1}$ des Rohrbogens $i = i_{|}$ bestimmt und der die Rohrstrecke repräsentierende Polygonzug entsprechend abgeändert. Die mit der KKF-Konstruktion gefundenen Positionen müssen wiederum, entsprechend den Zwangsbedingungen (4.8)-(4.10), welche den Platzbedarf der Rohrbögen berücksichtigen, gefiltert werden. Auch für diese Regel gilt die Einschränkung, dass, auf Grund sich gegenseitig widersprechender Zwangsbedingungen, der Regelaufruf auch erfolglos sein kann.

Winkeländerungen der Randbögen $i_{|} = 1$ beziehungsweise $i_{|} = N - 2$ können mit der KKF-Konstruktion bewerkstelligt werden, indem man während der Regelausführung "virtuelle" benachbarte Rohrbögen $i_{\leftarrow} = 0$ beziehungsweise $i_{\rightarrow} = N - 1$ mit $\bar{\alpha}_{i_{\leftarrow}} = \bar{\alpha}_{i_{\leftarrow}} = 0$ einführt. In diesem Falle reduziert sich der entsprechende Kegel der KKF-Konstruktion auf eine Linie.

Wie bei Regel B könnten in einer alternativen Regelformulierung im Rahmen der Anwendung der KKF-Konstruktion auch gleichzeitig die Bogenwinkel der benachbarten Bögen i_{\leftarrow} und i_{\rightarrow} unter Verwendung der KKF-Konstruktion geändert werden. Es kann wiederum zu sich wider-

sprechenden Zwangsbedingungen kommen, so dass mit Regel *C* keine neue Rohrbogenposition $\vec{P}'_{i+1} \neq \vec{P}_{i+1}$ bestimmt werden kann und somit der Rohrbogen i' unverändert bleibt.

Regel D: Rohrbogen entfernen Für die Entfernung eines Rohrbogens aus der Rohrstrecke wird ebenfalls die KKF-Konstruktion verwendet. Es wird wiederum von einer eingefrorenen Rohrstrecke ohne Bewegung der Rohrbögen im Starrkörpersimulator ausgegangen. Bei Entfernung des Rohrbogens i_l wird die KKF-Konstruktion auf das Rohrstreckensegment $\overline{P_{i_{\leftarrow}+1}P_{i_l+1}P_{i_{\rightarrow}+1}}$ unter Beibehaltung der Positionen $\vec{P}_{i_{\leftarrow}+1}$ und $\vec{P}_{i_{\rightarrow}+1}$ angewandt. Als Bogenwinkel werden $\bar{\alpha}'_{i_{\leftarrow}} = \bar{\alpha}_{i_{\leftarrow}}$, $\bar{\alpha}'_{i_l} = 0$ und $\bar{\alpha}'_{i_{\rightarrow}}$ vorgegeben.

Der rotierte Fasskreisbogen der KKF-Konstruktion reduziert sich durch die Wahl von $\bar{\alpha}'_{i_l} = 0$ auf eine Linie. Daraus folgt, dass die gegebenenfalls gefundene neue Zwischenposition \vec{P}'_{i+1} auf der Strecke $\overline{P_{i_{\leftarrow}+1}P_{i_{\rightarrow}+1}}$ zu liegen kommt und damit anschließend vom Polygonzug entfernt werden kann.

Entsprechend den Anwendungen der KKF-Konstruktion in den vorangegangenen Regeln kann es zu sich widersprechenden Zwangsbedingungen kommen. Damit ist eine Entfernung des Rohrbogens unter Beibehaltung der übrigen Bogenwinkel nicht garantiert. Sie hängt vielmehr von der konkreten Streckenkonfiguration zum Zeitpunkt der Regelausführung ab. Das Entfernen von Randbögen $i_l = 1$ beziehungsweise $i_l = N - 2$ wird wie in Regel *C* behandelt. Ebenfalls analog zu den Regeln *B* und *C* könnte mit der angewandten KKF-Konstruktion, in einer alternativen Regelformulierung, eine Gleichzeitige Modifikation der Bogenwinkel der zum entfernenden Rohrbogen angrenzenden Rohrbögen i_{\leftarrow} und i_{\rightarrow} vollzogen werden.

Regel E: Verschiebung der Rohrbögen Regel *E* stellt im Vergleich der vorangegangenen Regeln eine reine parametrischen Manipulation der Rohrstrecke dar. Die Zahl der Rohrbögen $N - 2$ sowie die Bogenwinkel $\bar{\alpha}_i$ bleiben unverändert. Es findet lediglich eine mit den Zwangsbedingungen verträgliche Positionsänderung der Rohrbögen statt. Im Starrkörpersimulator wird dazu die Kinematik der Rohrstrecke unter den gegebenen Zwangsbedingungen (Dreh-Schiebe-Gelenke, Kollisionen zwischen Platzhalter-Kugeln) simuliert. Die die Rohrstrecke repräsentierenden Elemente der Starrkörpersimulation müssen mit einer Masse initialisiert werden, da die Simulation der Starrkörperbewegungen auf einem physikalischen Starrkörpermodell beruht. Diese Masse kann für den vorliegenden Anwendungsfall beliebig²² gewählt werden, da es nicht auf die physikalische Repräsentation der Starrkörper-Dynamik der Rohrstrecke ankommt, sondern lediglich die Manipulation der Rohrstrecke unter den gegebenen Zwangsbedingungen interessiert.

Die Verschiebung der Rohrbögen wird im Starrkörpersimulator über das Aufbringen einer entsprechenden Kraft \vec{F}_i auf den Rohrbogen i repräsentierenden Körper beziehungsweise Kugel realisiert. Auf Grund der durch die Gelenke eingebrachten und gekoppelten Zwangsbedingungen führt dies im Allgemeinen nicht auf eine isolierte Bewegung des jeweiligen Rohrbogens, sondern zu einer gekoppelten Bewegung der benachbarten Bögen. Für die weitere Anwendung soll davon ausgegangen werden, dass in der betrachteten Regel *E* auch auf mehrere Rohrbögen gleichzeitig Kräfte aufgebracht werden können.

²²Im idealen Falle wird die Masse optimal im Sinne der numerischen Stabilität der sich ergebenden Bewegungsgleichungen gewählt. Bei zu kleinen Massen ergeben sich sehr große Schrittweiten, da sich das System sprunghaft verhält. Dies macht eine sehr feine Diskretisierung der Zeit nötig und führt so zu einer hohen Zahl an Simulationsschritten. Bei großen Massen verhält sich die Rohrstrecke im Simulator sehr träge und es braucht sehr viele Simulationsschritte zur Verschiebung der Rohrstrecke. Hierzu sei auch das Handbuch des verwendeten Starrkörpersimulationsprogrammes empfohlen [177].

4.1.4. Berücksichtigung weiterer Zwangsbedingungen

Die bisher beschriebene Umsetzung der Verrohrungssprache im Starrkörpersimulator berücksichtigt lediglich die Einhaltung der Start-/Endpositionen und Richtungen $(\vec{P}_S, \vec{P}_E, \vec{D}_S, \vec{D}_E)$, sowie die vorgegebenen Bogenwinkel $\bar{\alpha}_i$. Für die Synthese von Rohrstrecken im ganzheitlichen Entwurfsproblem müssen aber noch weitere Randbedingungen berücksichtigt werden.

Der zur Verfügung stehende Einbauraum sei hier an erster Stelle genannt. Der Bauraum ist in aller Regel durch zuvor schon eingebaute Systeme und Komponenten beziehungsweise durch im Gesamtentwurf fest vorgegebene Anforderungen eingeschränkt und soll als 3D (CAD-)Modell vorliegen. Die Bauraum- sowie die Start-/Endpositions- und Richtungs-Zwangsbedingungen stellen, zusammen mit den vorgegebenen Rohrbogenkonfigurationen (I, R) , harte Zwangsbedingungen [49] der Verrohrungssprache dar. Diese müssen zwingend eingehalten werden. Weitere harte Randbedingungen, wie zum Beispiel Gebiete des Bauraumes, welche für die Verrohrung aus nicht-geometrischen Gründen gesperrt sind, können über eine entsprechende Modifikation des Bauraummodells berücksichtigt werden. Gegenüber harten Zwangsbedingungen müssen weiche Randbedingungen [49] nicht zwingend, sondern sollen möglichst eingehalten werden. Diese weichen Zwangsbedingungen werden bevorzugt in Form von Kostenfunktionen abgebildet [49].

Als erste Möglichkeit der Berücksichtigung der Bauraumzwangsbedingungen wurde in der vorliegenden Arbeit untersucht, die Bauraumgrenzen (Oberflächenmodell des Bauraumes) direkt in die Kollisionsbehandlung des Starrkörpersimulators zu integrieren. Der verwendete Starrkörpersimulator [177] ist in der Lage, triangulierte Oberflächen (STL-Format) als Hindernisse in der Kollisionsbehandlung zu berücksichtigen. Damit ist es theoretisch möglich, die Rohrsuche direkt im Bauraum durchzuführen. Dieser Ansatz wurde aber auf Grund der unzureichenden Performance der Kollisionsbehandlung der im Allgemeinen recht komplexen Bauraumgrenzen wieder verworfen. Dieser direkte Ansatz der Behandlung der Bauraumgrenzen soll aber nicht unerwähnt bleiben, da er in der Zukunft, mit weiter steigenden Rechenleistungen, vorteilhaft eingesetzt werden kann.

Der direkte Ansatz mit Kollisionsbehandlung der Bauraumgrenzen im Starrkörpersimulator weist in der Initialisierung der Rohrstrecken eine weitere Schwierigkeit auf: Es muss sichergestellt werden, dass die Bögen der initialen Rohrstrecke innerhalb des Bauraumes liegen. Bei eingeschalteter Kollisionsbehandlung mit den Bauraumgrenzen können sich die Rohrbögen, welche nach der Initialisierung außerhalb des Bauraumes liegen, nicht durch die Bauraumgrenzen bewegen um in den Bauraum zurückzukehren.

Im Rahmen dieser Arbeit soll das Problem der Berücksichtigung des Einbauraumes in der Verrohrungssprache über den Einsatz eines sogenannten Referenzpfades gelöst werden. Dazu wird der zur Verfügung stehende Bauraum in Zellen ω aufgeteilt. Der so diskretisierte Bauraum Ω bilde den Graphen $\Omega(P, E)$ mit Knoten P und Kanten E . Die Knotenmenge P wird aus den N Ortsvektoren der Zellmittelpunkte $\vec{\omega}_i \in P = \{\vec{\omega}_1, \dots, \vec{\omega}_i, \dots, \vec{\omega}_N\}$ gebildet. Die Nachbarschaftsbeziehungen der Zellen wird in den Kanten $e_k \in E = \{e_1, \dots, e_k, \dots, e_M\}$ gespeichert: Direkt benachbarte Zellen ω_i und ω_j , die im räumlichen Fall eine gemeinsame Grenzfläche besitzen, sollen eine Kante $e_k = e_k(\vec{\omega}_i, \vec{\omega}_j) \in E$ des Graphen bilden.

Auf dem Graph Ω kann nun ein Pfadsuchalgorithmus (siehe Kapitel 2.4.1 und 2.4.2) ausgeführt werden. Dieser findet auf dem Graph Ω des diskretisierten Bauraumes einen Pfad, in Form einer Kantenliste $\bar{E} = \{e_1(\vec{\omega}_S, \vec{\omega}_i), \dots, e_k, \dots, e_{|E|}(\vec{\omega}_j, \vec{\omega}_E)\}$, zwischen den Zellen ω_S und ω_E , welche die Start- und Endposition, $\vec{P}_S \in \omega_S$ beziehungsweise $\vec{P}_E \in \omega_E$, der Verrohrung enthalten.

Die Verwendung eines Pfadsuchalgorithmus auf dem diskretisierten Bauraum, zur Auffindung eines Referenzpfades, kann zur Berücksichtigung weicher Zwangsbedingungen verwendet werden. Die Kanten $e_k \in E$ des Graphen Ω können mit Kostenwerten $|e_k|$,

$$|e_k| = F(e_k) = f(\vec{x}_k) + f(e_k), \quad (4.16)$$

die sich aus ortsabhängigen Kosten $f(\vec{x}_k)$ und kantenbezogenen Kosten $f(e_k)$ zusammensetzt, versehen werden und bilden so einen gewichteten Graph. Der einer Kante $e_k(\vec{\omega}_i, \vec{\omega}_j)$ zugeordnete Ortsvektor \vec{x}_k soll als Mittelpunkt der Zellzentren $\vec{\omega}_i$ und $\vec{\omega}_j$ der an der Kante beteiligten Zellen ω_i und ω_j bestimmt werden:

$$\vec{x}_k = \frac{1}{2} \cdot (\vec{\omega}_i + \vec{\omega}_j). \quad (4.17)$$

Die Suche eines Pfades mit minimalem Gewicht auf dem Graph stellt das in Kapitel 2.4.1 aufgeführte Shortest Path Problem dar. Die entsprechenden Algorithmen können dann zur Bestimmung eines optimalen Pfades, $\bar{E} = \{e_S(\vec{\omega}_S, \vec{\omega}_i), \dots, e_j, \dots, e_S(\vec{\omega}_k, \vec{\omega}_E)\}$, mit minimaler Summe der Kantengewichte,

$$\bar{F} = \sum_{e_k \in \bar{E}} |e_k| = \sum_{e_k \in \bar{E}} F(e_k) = \min, \quad (4.18)$$

verwendet werden. Dieser Pfad bildet einen Polygonzug im Bauraum, dessen Aufhängepunkte aus den entsprechenden Zellzentren, $\vec{\omega}_i$ mit $\vec{\omega}_i \in e_k \in \bar{E}$, gebildet werden.

Typische ortsabhängige Zwangsbedingungen, welche in die ortsabhängige Kostenfunktion $f(\vec{x}_k)$ eingehen, sind beispielsweise:

- Abstandsregeln zu anderen System-Komponenten im Bauraum (z.B. Aufhängepunkte)
- Abstandsregeln zu den Bauraumgrenzen (wandferne / wandnahe Verlegung der Rohrstrecke)
- Minimierung / Maximierung von Feldeinflüssen (z.B. Elektromagnetische Verträglichkeit)
- Positionierung relativ zu anderen Rohrstrecken (z.B. zur parallelen Verlegung von Rohrstrecken)

Dagegen geht zum Beispiel die Pfadlänge in Form der Kantenlängen in die kantenbezogene Kostenfunktion $f(e_k)$ ein. Biegungsbezogene Bedingungen gehen direkt über die Rohrbogenkonfiguration (I, R) und die Anzahl der Rohrbögen in die Rohrstrecke ein. Durch eine Gewichtung der einzelnen Zwangsbedingungen in den Nutzenfunktionen kann eine gegenseitige Priorisierung der Ziele vollzogen werden.

Der so gefundene Referenzpfad, welcher wie beschrieben zur Berücksichtigung der weichen Randbedingungen verwendet werden kann, liegt per Definition im zur Verfügung stehenden Bauraum. Dieser soll während der Regelausführung im Produktionssystem möglichst gut mit der unter Verwendung der vorgegebenen Rohrbögen generierten Rohrstrecke angenähert werden. Dazu wird in der Synthese der Rohrstrecke versucht, den Abstand der Rohrstrecke zum Referenzpfad zu minimieren.

Wie sich im weiteren Verlauf der Arbeit zeigen wird, liefert die Verwendung des Referenzpfades akzeptable Ergebnisse für die betrachteten Entwürfe. Mit einer abschließenden Überprüfung, ob die gefundene Rohrstrecke die Bauraumgrenzen einhält, kann die Validität der Verrohrung sichergestellt werden. Die Verwendung des Referenzpfades stellt damit eine im Rahmen dieser

Arbeit ausreichende Alternative zur direkten Berücksichtigung der Bauraumgrenzen in der Kollisionsbehandlung des Starrkörpersimulators dar.

Die Anziehung der im Starrkörpersimulator abgebildeten Rohrstrecke an den Referenzpfad soll über einen kraftgesteuerten Ansatz erfolgen. Das Aufbringen der Kräfte auf die Rohrstrecke im Starrkörpersimulator wird mit Hilfe von Regel E bewerkstelligt, welche, wie beschrieben, auf mehrere Rohrbögen gleichzeitig Kräfte zur Verschiebung wirken lassen kann.

Die Kraft \vec{F}_k zur Verschiebung des Rohrbogens k wird im Starrkörpersimulator jeweils auf die den Rohrbogen darstellende Kugel mit der zugeordneten Position $\vec{P}_{i=k+1}$ aufgebracht. Für die Bestimmung des, auf den jeweiligen Rohrbogen aufzubringenden, Kraftvektors wird zunächst der Polygonzug der Rohrstrecke Π (siehe Abschnitt 4.1.2) näher betrachtet: Die Punkte $\vec{P}_{i=2=k+1}$ bis $\vec{P}_{i=N-1}$ des Polygonzuges stellen die Positionen der Rohrbögen dar. Deren relative Position $p_{i=k+1}$ entlang des Polygonzuges ergibt sich aus der Gesamtlänge L des Polygonzuges,

$$L = \sum_{i=1}^{N-1} \sqrt{(\vec{P}_i - \vec{P}_{i+1})^2}, \quad (4.19)$$

zu:

$$p_{i=k+1} = \frac{1}{L} \cdot \sum_{j=1}^{i-1} \sqrt{(\vec{P}_j - \vec{P}_{j+1})^2}. \quad (4.20)$$

Der mit dem Pfadsuchalgorithmus (Abschnitt 4.1.4) gewonnene Referenzpfad \vec{E} , der alternativ zur Darstellung als Kantenliste auch äquivalent als Punktliste $\vec{E} = \{\vec{\omega}_S, \dots, \vec{\omega}_j, \dots, \vec{\omega}_E\}$ der Zellzentren geschrieben werden kann, stellt ebenfalls einen Polygonzug dar. Die Länge dieses Polygonzuges kann entsprechend mit Gleichung (4.19) bestimmt werden.

Auf dem Polygonzug des Referenzpfades werden nun den Positionen $\vec{P}_{i=k+1}$ der Rohrbögen k der Rohrstrecke entsprechende Punkte $\vec{\omega}'_{i=k+1}$ auf dem Referenzpfad mit²³

$$p_{i=k+1} = p_{\omega_{i=k+1}} \quad (4.21)$$

zugeordnet, wobei $p_{\omega_{i=k+1}}$ entsprechend Gleichung (4.20) die relative Positionierung des Rohrbogens k mit Position $\omega'_{i=k+1}$ auf dem Referenzpfad darstellt. Damit besitzen die zwischen der Rohrstrecke und dem Referenzpfad einander zugeordneten Positionen $\vec{P}_{i=k+1}$ und $\vec{\omega}'_{i=k+1}$ eines Rohrbogens k dieselbe relative Position entlang ihrer jeweiligen Strecke.

Daraus ergibt sich die Kraft \vec{F}_k auf den Rohrbogen k mit Position $\vec{P}_{i=k+1}$ zu:

$$\vec{F}_k \propto \vec{\omega}'_{i=k+1} - \vec{P}_{i=k+1}. \quad (4.22)$$

Das gleichzeitige Aufbringen der Kraft \vec{F}_k auf allen Rohrbögen $1 \leq k \leq N - 2$ führt zu einer Anziehung und Annäherung der Rohrstrecke Π an den stationären Referenzpfad \vec{E} .

4.1.5. Produktionssystem der Verrohrungssprache

In diesem Abschnitt sollen zwei mögliche Produktionssysteme auf Basis des oben definierten Regelwerks zur regelbasierten Synthese von Rohrstrecken dargestellt werden: Die erste Variante basiert auf dem schon beschriebenen stochastischen Optimierungsprinzip der Simulierten Abkühlung. Die zweite Variante soll eine rein parametrische Optimierung, unter Verwendung des Referenzpfades und unter Beibehaltung der Rohrbögen der Rohrstrecke, darstellen.

²³Indexierung i bezüglich der Polygonpunkte der Rohrstrecke mit $i = k + 1$ und der Indexierung k der Rohrbögen.

Synthese der Rohrstrecke mit Simulierter Abkühlung Das Produktionssystem unter Verwendung der Simulierten Abkühlung folgt der "reinen" Entwurfssprachenlehre wie sie in den Abschnitten 2.3 und 4.1.1 wiedergegeben wurde. Dieses Produktionssystem soll alle in obigem Kapitel 4.1.3 formulierten Regeln *A-E* umfassen.

Bei der Verwendung des Referenzpfades in der Synthese der Rohrstrecke, soll die Verschiebungsregel *E*, wie in Abschnitt 4.1.4 gezeigt, zur Anziehung der Rohrstrecke an den Referenzpfad verwendet werden. Die Reihenfolge Regelausführung im Produktionssystem soll, analog zu [187], im Optimierungsschema der Simulierten Abkühlung (Kapitel 2.5.2) bestimmt werden.

Als erste Regel des Produktionssystems wird das Axiom (Regel *A*) einmalig aufgerufen und eine initiale Rohrstrecke synthetisiert. Dann wird die Güte W_t der aktuellen Rohrstrecke bestimmt. Anschließend wird in jedem Optimierungsschritt Δt eine zufällig aus *B-E* ausgewählte Regel, mit zufälliger Parametrisierung, aufgerufen und die Güte $W_{t+\Delta t}$ des so modifizierten Systems berechnet. Der neue Zustand der Rohrstrecke wird auf jeden Fall akzeptiert, wenn er eine höhere Güte aufweist als die Rohrstrecke des letzten Schrittes:

$$W_{t+\Delta t} \geq W_t. \quad (4.23)$$

Eine Verschlechterung der Güte der Rohrstrecke um

$$\Delta W = W_t - W_{t+\Delta t} \geq 0 \quad (4.24)$$

wird mit einer Wahrscheinlichkeit von

$$P_{akzeptiert} = e^{\frac{-\Delta W}{\Delta T}} \quad (4.25)$$

akzeptiert und die geänderte Rohrstrecke als aktuelle Konfiguration übernommen. Der Parameter T wird in Anlehnung an die Boltzmann-Statistik als Temperatur bezeichnet und fällt im Laufe der Optimierung für steigende t monoton ab. Für hohe Werte T , am Anfang der Optimierung (kleine Werte von t), werden so auch Zustände mit erheblicher Verschlechterung der Güte der Rohrstrecke akzeptiert.

Dies führt zu einem großräumigen Absuchen des Entwurfsraumes der Rohrstrecke. Für kleine Werte von T , bei fortgeschrittenem Fortgang der Optimierung, werden praktisch nur die Zustände akzeptiert, welche eine Verbesserung der Güte bewirken. Dies führt zur Annäherung an ein Minimum der Gütefunktion. Der konkrete Abkühlungsplan für die Synthese ist in der Darstellung der Beispiele im folgenden Kapitel 4.2.1 dargestellt.

Die Gütefunktion W kann sich direkt aus den Bewertungsgrößen der jeweiligen Rohrstrecke bezüglich der Einhaltung der Randbedingungen zusammensetzen: Wandabstand, Anzahl der Rohrbögen, Summe Rohrbogenwinkel, Bauraumverletzung,... Alternativ kann auch der Abstand zwischen Rohrstrecke und Referenzpfad, welcher wie oben beschrieben die (weichen) Randbedingungen in sich vereinen kann, als Güte der Optimierung eingesetzt werden. Diese muss in der Optimierung mit der Simulierten Abkühlung minimiert werden. Eine solche Minimierung kann praktisch über die Maximierung des Kehrwertes der Gütefunktion erreicht werden.

Parametrische Optimierung und Reinitialisierung der Rohrbogenkonfiguration Die zweite Möglichkeit eines Produktionssystems zur Synthese von Rohrstrecken, die in dieser Arbeit betrachtet werden soll, beschränkt sich auf die Verwendung der Regeln *A* und *E*. Letztere Regel wird, wie in Abschnitt 4.1.4 beschrieben, zur Anziehung der Rohrstrecke an

einen Referenzpfad, welcher den Verlauf der Rohrstrecke entsprechend der Randbedingungen vorgibt, oder zur zufälligen Verschiebung einzelner Rohrbögen verwendet.

Im Gegensatz zur vorher beschriebenen Optimierung der Rohrstrecke mit Simulierter Abkühlung, welche eine regelbasierte Modifikation der Rohrbogenkonfiguration umfasst, findet im alternativen Produktionssystem eine Neuinitialisierung der Rohrstrecke in jedem Optimierungsschritt statt. Dabei wird durch den Aufruf des Axioms (Regel *A* jeweils mit einer anderen Zahl von Rohrbögen bzw. modifizierten Rohrbogenwinkeln reinitialisiert. Anschließend wird ohne weitere Variation der Rohrbogenzahl und der Rohrbogenwinkel, bei Verwendung eines Referenzpfades, die Verschiebungsregel *E* zur Anziehung der Rohrstrecke an den Referenzpfad aufgerufen. Die so für jede untersuchte Rohrbogenkonfiguration gefundene, an den Referenzpfad angenäherte, Rohrstrecke kann dann bewertet werden. Damit lässt sich die Konfiguration von Rohrbögen bestimmen, welche eine optimale Annäherung an den Referenzpfad verspricht.

Die Verschiebungsregel *E* kann aber auch alternativ zur Verschiebung von einzelnen Rohrbögen im Rahmen einer parametrischen Optimierung unter direkter Bewertung der jeweiligen Rohrstrecke, ohne Einsatz eines Referenzpfades, verwendet werden.

Diese Herangehensweise der Neuinitialisierung mit parametrischer Optimierung einer Konfiguration erlaubt, im Sinne des bereits erwähnten Prinzips des einfachen Entwurfes (KISS-Prinzip [148]), eine automatisierte und systematische Prüfung von Rohrbogenkonfigurationen der Rohrstrecke. Dabei liegt es nahe, mit einer Konfiguration mit minimaler Rohrbogenzahl, welche die Bedingung (4.1) bezüglich der mindestens notwendigen Winkelsumme der Rohrbögen erfüllt, zu beginnen und danach die Anzahl der Rohrbögen und damit die Komplexität der untersuchten Rohrstrecke schrittweise zu erhöhen.

4.2. Anwendungsbeispiele

Dieses Kapitel soll Ergebnisse der im vorherigen Abschnitt 4.1.5 definierten Produktionssysteme zeigen. Für die Visualisierung und die Weiterverwendung der gefundenen Verrohrungen wird die Verrohrungssprache mit einer CAD-Schnittstelle auf Basis der frei erhältlichen und quell-offenen CAD-Bibliothek Open Cascade [136] versehen. Diese Schnittstelle erlaubt den Export der erzeugten Geometrien in den Austauschformaten STEP [94] und IGES [191].

4.2.1. Synthese mit Simulierter Abkühlung

Für die Simulierte Abkühlung soll ein vereinfachter Abkühlungsplan (Annealing Schedule) verwendet werden. Dies stellt eine Abweichung zum Abkühlungsplan der Erstveröffentlichung [103] dar. Dieser sieht vor, dass die Temperatur so lange konstant gehalten wird bis das System einen stationären Zustand in der gegebenen Temperatur erreicht hat. Dies ist aber auf Grund der nicht unerheblichen Berechnungsdauer eines Optimierungsschrittes, der sich für die Optimierung einer Rohrstrecke im zweistelligen Sekundenbereich bewegt, mit den für diese Arbeit zur Verfügung stehenden Rechenleistungen nicht umsetzbar. Das Erreichen eines sich im statistischen Gleichgewicht befindlichen Zustandes benötigt im Allgemeinen eine sehr große Zahl an synthetisierten Konfigurationen.

Als Startwert des Temperaturparameters T soll für die Beispiele jeweils ein Wert von $T = 1000$ gewählt werden. Der vereinfachte Abkühlungsplan soll nach jedem akzeptierten Optimierungsschritt (siehe Gleichung (4.25)) eine relative Reduktion der Temperatur um einen vorgegebenen Prozentsatz von $\delta T = 1\%$ vorsehen.

Prinzipbeispiel: Würfel Zur Demonstration der Funktion der Verrohrungssprache auf Basis der Simulierten Abkühlung wird zunächst ein einfaches Beispiel, die diagonale Verrohrung durch einen Würfel analog zur Arbeit [187], gewählt. Dabei soll die Synthese einer einzelnen Rohrstrecke sowie die Synthese eines Ensembles von drei Rohrstrecken betrachtet werden. Die Start- und Endrichtungen sind parallel, entlang einer der Achsen der Box, ausgerichtet. Für die Verrohrung der drei Rohrstrecke werden die Start- und Endpositionen der Strecken entlang einer zur Start- und Endrichtung rechtwinkligen Achse ausgerichtet. Die Punkte sollen als zusätzliche Schwierigkeit so gewählt werden, dass sich die beiden außen liegenden Strecken überkreuzen müssen.

Es soll zunächst eine Rohrstrecke, welche entsprechend Kapitel 4.1.4 an einen Referenzpfad gezogen soll, synthetisiert werden. Die Menge der zugelassenen Bogenwinkel sei $\bar{\alpha} = \{30^\circ, 45^\circ, 60^\circ\}$ und der Biegeradius soll dem Rohrdurchmesser entsprechen: $R = D$. Der Abstand $d(\text{Rohrstrecke}, t, \text{Referenzpfad})$ der Rohrstrecke I' zum Referenzpfad im Zeitschritt t stellt die zu minimierende Gütefunktion der Optimierung dar (entspricht der Maximierung des Kehrwertes):

$$W_t(I') = d(\text{Rohrstrecke}, t, \text{Referenzpfad}) \quad (4.26)$$

Die Funktion $d(\text{Rohrstrecke}, t, \text{Referenzpfad})$ berechnet den Abstand der Rohrstreckenkonfiguration im Zeitschritt t vom stationären Referenzpfad. Durch die Normierung des Abstandes, beispielsweise auf den Rohrdurchmesser, kann eine dimensionslose Bewertungsgröße gewonnen werden. Für das vorliegende Beispiel wird für die Berechnung des Abstandes die Rohrstrecke und der Referenzpfad in eine identische Zahl äquidistanter Punkte von der Start- zur Endposition zerlegt. Der Abstand jeweils identisch indexierter Punkte der beiden Strecken wurde dann zur gesuchten Größe, dem Abstand der beiden Strecken, aufsummiert.

Darüber hinaus soll eine zweite Rohrstrecke, ohne Verwendung eines Referenzpfades, mit minimaler Gesamtbiegung bestimmt werden. Dies soll mit einer alternativen Gütefunktion, welche aus der Winkelsumme der Rohrbögen k der Rohrstrecke I' zu jedem Zeitschritt t der Simulierten Abkühlung gebildet wird, umgesetzt werden.

$$W_t(I') = \sum_k \bar{\alpha}_k \quad (4.27)$$

Diese Gütefunktion ist per Definition eine dimensionslose Bewertungsgröße. Die Menge der zugelassenen Bogenwinkel bei Verwendung dieser Gütefunktion sei $\bar{\alpha} = \{15^\circ, 30^\circ\}$, wiederum mit $R = D$.

Das Verhindern gegenseitiger Durchdringung der Rohrstrecken kann prinzipiell durch die Kollisionsbehandlung des Starrkörpersimulators sichergestellt werden. Während der simultanen Evolution und Synthese mehrerer Rohrstrecken in einem gemeinsamen Bauraum soll aber auch eine gegenseitige Durchdringung beziehungsweise ein gegenseitiges Durchtunneln der Rohrstrecken ermöglicht werden, so dass sich beispielsweise knotenähnliche Blockaden zweier oder mehrerer Rohrstrecken auflösen können.

Die Kollisionsbehandlung zur Verhinderung von Durchdringungen soll aus diesem Grund zufällig, nur in jedem zweiten Aufruf der im Starrkörpersimulator ausgeführten Regel E , eingeschaltet sein. Dies bedeutet aber, dass es nach der Ausführung der Regel E mit abgeschalteter Kollisionsbehandlung zu gegenseitiger Durchdringung der Rohrstrecken kommen kann. Ebenso kann es nach der Ausführung der Regeln B - D , die eine Manipulation der Rohrbogenkonfiguration mit Hilfe der KKF-Konstruktion vollzieht, welche selbst keine Durchdringungsprüfung enthält, zu Durchdringungen zwischen den Rohrstrecken oder einer Rohrstrecke mit sich selbst kommen.

Aus diesem Grund wird die gegenseitige Durchdringung der Rohrstrecken als weitere Bewertungsgröße in die Gütefunktion eingeführt. Damit kann die Durchdringung der Rohrstrecken über die Bewertung der Rohrstrecke in der Simulierten Abkühlung berücksichtigt und so herausgefiltert werden.

Dazu werden die Gütefunktionen (4.26) beziehungsweise (4.27) der Rohrstrecke I' mit einer von der Durchdringung D_t der Strecken abhängigen Gütefunktion $w(D_t)$ multipliziert:

$$W'_t(I') = W_t(I') \cdot w(\bar{D}_t(I')). \quad (4.28)$$

Die Gütefunktion $w(\bar{D}_t(I'))$ soll für durchdringungsfreie Rohrstrecken den Wert eins haben. Für eine Durchdringung der Rohrstrecken sollen größere Werte zurückgegeben werden, so dass in diesem Fall die zu minimierenden Gütefunktionen erhöht werden. Die gegenseitige Durchdringung der Rohrstrecken wird in der vorliegenden Arbeit aus den einzelnen Durchdringungen der Geradensegmente des Polygonzug-Ersatzmodelles (siehe Kapitel 4.1.2) der Rohrstrecke bestimmt. Die Durchdringung $\bar{d}(i, j)$ des Segments $i \in I'$ aus Rohrstrecke I' und des Segments $j \in J$ aus Rohrstrecke J mit $I' \neq J$ berechnet sich näherungsweise mit Hilfe des geometrischen Abstandes $d(i, j)$ der beiden Geradenstücke und den Rohrdurchmessern $D_{I'}$ und D_J der jeweiligen Rohrstrecken zu:

$$\bar{d}(i, j) = 0.0 \quad , \text{ für } d(i, j) \geq 0.5 \cdot (D_{I'} + D_J), \quad (4.29)$$

$$\bar{d}(i, j) = 0.5 \cdot (D_{I'} + D_J) - d(i, j) \quad , \text{ für } d(i, j) < 0.5 \cdot (D_{I'} + D_J). \quad (4.30)$$

Dazu wird über alle möglichen Kombinationen der im betrachteten Bauraum simultan zu synthetisierenden Rohrstrecken I' und J sowie der Polygonzugsegmente i und j der jeweiligen Strecken summiert. Die Berechnung der Durchdringung von Segmenten der gleichen Linie, $I' = J$, sei ausgeschlossen. Eine Normierung der Durchdringung $\bar{d}(i, j)$ auf den durchdringungsfreien Abstand $0.5 \cdot (D_{I'} + D_J)$ der Rohrstrecken I' und J liefert eine dimensionslose Bewertungsgröße der Durchdringung. Damit ergibt sich für Linie I' eine Durchdringung von

$$\bar{D}(I') = \sum_{J \neq I'} \sum_{i \in I'} \sum_{j \in J} \frac{\bar{d}(i, j)}{0.5 \cdot (D_{I'} + D_J)} \quad (4.31)$$

mit den anderen Rohrstrecken J .

Die Gütefunktion der Durchdringung soll für Durchdringungswerte von $\bar{D}(I') = 0$ den Wert von eins annehmen und in der Optimierung ein großes Gewicht besitzen, da eine Durchdringung nicht akzeptabel ist. Dafür bietet sich eine exponentielle Funktion mit einem Parameter C , zur Regulierung des Gewichtes der Durchdringung innerhalb der Gütefunktion, an:

$$w(\bar{D}_t(I')) = e^{C \cdot \bar{D}_t(I')} \text{ mit } C > 0. \quad (4.32)$$

Die Beispiele mit den jeweiligen Gütefunktionen (4.26) und (4.27) sollen unter Verwendung der in Kapitel 4.1.1 definierten Regeln berechnet werden. Als erste wird Regel A zur Initialisierung einer ersten Rohrstrecke ausgeführt. Die Regeln B - D werden wie in Kapitel 4.1.3 beschrieben angewendet. Die Regel E wird in Abschnitt 4.1.3 in zwei Varianten angewendet: Regel E_1 soll auf einen zufällig gewählten Rohrbogen eine zufällige Kraft aufbringen. Regel E_2 soll entsprechend Gleichung (4.22) aus Abschnitt 4.1.4 implementiert sein und zur Anziehung der Rohrstrecke an den Referenzpfad dienen.

Die im Starrkörpersimulator ausgeführten Regeln E_1 und E_2 werden wie oben dargestellt in jeweils zwei zufällig, mit gleichen Wahrscheinlichkeiten, gewählten Varianten aufgerufen: Eine

Variante mit Kollision der Rohrstrecken. In der zweiten Variante ist die gegenseitige Kollisionsdetektion der Rohrstrecken ausgeschaltet, so dass eine Umordnung der Rohrstrecken möglich wird und ein gegenseitiges Blockieren der Strecken verhindert wird.

In jedem Optimierungsschritt soll eine Regel aus $\{B, C, D, E_1, E_2\}$ zufällig und mit gleicher Wahrscheinlichkeit aufgerufen werden. In der gleichzeitigen Optimierung mehrerer Strecken in einem Bauraum wird auf den Strecken jeweils die gleiche, zufällig gewählte Regel angewendet. Die Bewertung und Akzeptierung einer modifizierten Strecke soll für jede Strecke getrennt vollzogen werden. Die Einzelstrecken stehen aber, in den im Starrkörpersimulator ausgeführten Regelaufrufen (Regel A und E_x), im Falle eingeschalteter Kollisionsdetektion in direktem Kontakt miteinander.

Die linke Spalte in Abbildungen 4.12 zeigt das Ergebnis der Verrohrungssprache unter Verwendung der Gütefunktion (4.26) bei Verwendung eines Referenzpfades und Minimierung des Abstandes. Der eingesetzte Referenzpfad ist grün dargestellt.

Die rechte Spalte der Abbildung 4.12 zeigt das Ergebnis der Verrohrungssprache mit dem identischen Abkühlungsplan unter Verwendung der Gütefunktion (4.27) zur Minimierung der Gesamtbiegung. Diese kommt entsprechend Abschnitt 4.1.5 ohne Verwendung eines Referenzpfades aus. Hierbei stellt der grün eingezeichnete Pfad die direkte Verbindung der Eckpunkte des Würfels zur Orientierung dar.

In der Abbildung 4.12 oben ist jeweils die Synthese einer einzelnen Strecke dargestellt. Die untere Hälfte zeigt die Ergebnisse für die gleichzeitige Synthese von drei Rohrstrecken.

Die Ergebnisse der drei parallelen Rohrstrecken zeigen den Einfluss des Referenzpfades auf die Synthese: Die Kreuzung der äußeren Strecken findet im unteren Bereich des Würfels, im Bereich und entlang des querverlaufenden Referenzpfades, statt. Die Optimierungsvariante ohne Berücksichtigung des Referenzpfades, basierend auf Gütefunktion (4.27), führt auf Rohrstrecken mit reduzierter Bogenwinkelsumme und einer Kreuzung der Rohrstrecken in der Mitte des Würfels. Die Tabelle 5 zeigt eine Gegenüberstellung der Bogenwinkelsummen für die beiden betrachteten Gütefunktionen: Verwendung des Referenzpfades (Gütefunktion (4.26)) und Gütefunktion (4.27) zur Minimierung der Bogenwinkelsumme.

	Freier Referenzpfad Gütefunktion (4.26)	Minimierung Bogenwinkel Gütefunktion (4.27)
Einfache Rohrstrecke	450°	150°
Dreifache Rohrstrecke	675°	555°

Tabelle 5: Vergleich der Bogenwinkelsummen für die Ergebnisse der Verrohrungssprache dargestellt in Abbildung 4.12.

Die Abbildung 4.13 zeigt den Verlauf der Güte der akzeptierten Rohrstrecke gegen den Temperaturparameter für das einsträngige Beispiel unter Verwendung des Referenzpfades (Ergebnis aus Abbildung 4.12 links oben). Die typische Charakteristik der simulierten Abkühlung, die Hinnahme von Konfigurationen mit minderer Güte für hohe Werte des Temperaturparameters, ist gut zu erkennen. Im Verlauf der Optimierung nimmt der Betrag der akzeptierten Güteverschlechterung mit abnehmenden Temperaturparameter ab.

Die Bogenwinkelsumme ist, wie die Güte, für die jeweils aktuelle und damit akzeptierte Rohrstrecke aufgetragen. Diese beiden Größen stagnieren im Bereich kleiner werdender Temperaturen, was bedeutet, dass keine besseren Rohrstrecken synthetisiert werden konnten und die Wahrscheinlichkeit für die Akzeptierung von Rohrstrecken mit etwas schlechteren Güten auf

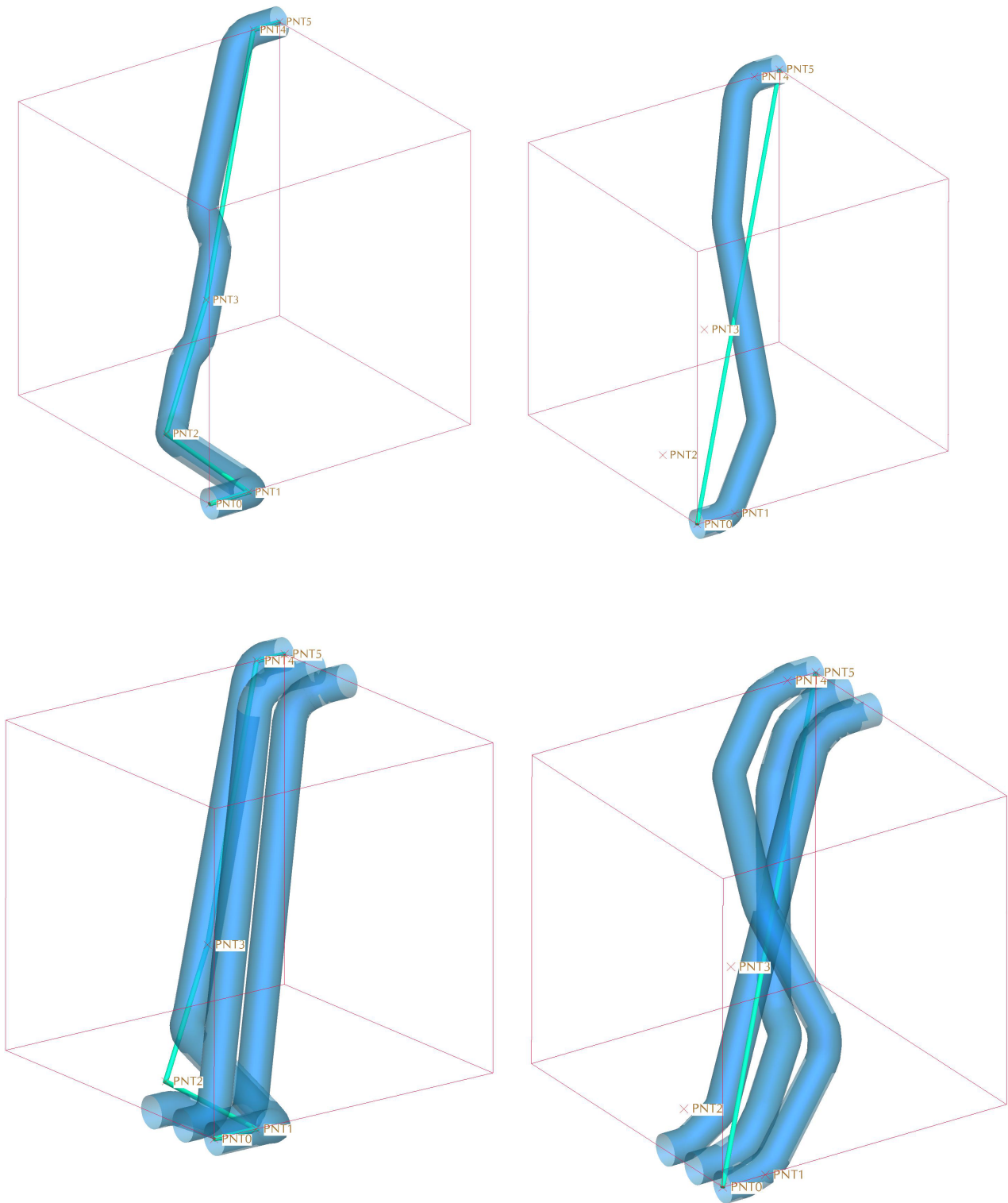


Abbildung 4.12: Prinzipdemonstration der simulierten Abkühlung der Verrohrungssprache. Linke Seite: Frei gewählter Referenzpfad (grün) mit Gütefunktion (4.26). Rechte Seite: Minimierung der Bogenwinkel mit Gütefunktion (4.27). Direkte Verbindung (türkis) und Stützpunkte des frei gewählten Referenzpfades zur Orientierung eingezeichnet. Oben: Verrohrung einer Strecke (blau). Unten: Gleichzeitige Verrohrung dreier Strecken (blau) mit Überkreuzung der Strecken.

Grund des niedrigen Temperaturparameters stark abnehmen. Für kleine Temperaturparameter werden damit praktisch nur noch Konfigurationen mit einer Verbesserung der Güte akzeptiert.

Das Absuchen verschiedener Konfigurationen, auch bei stagnierendem Wert der Gütefunktion, ist an dem schwankenden Verlauf der Bogenanzahl der jeweils synthetisierten und damit auch der nicht-akzeptierten Strecken in Abbildung 4.13 zu erkennen.

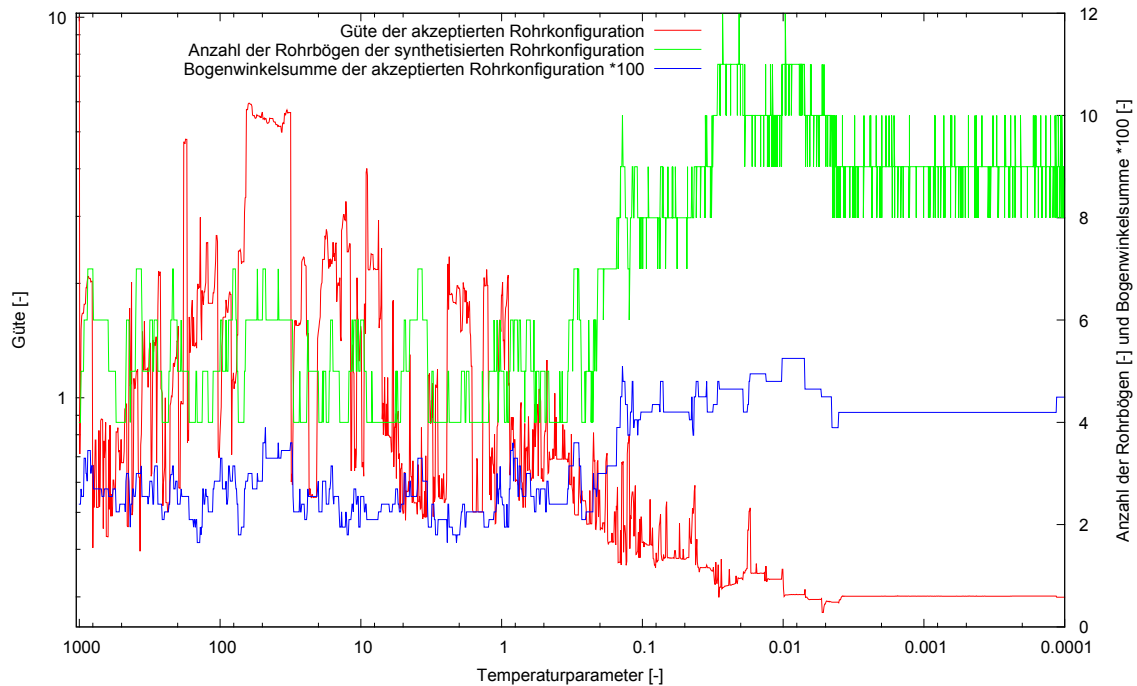


Abbildung 4.13: Verlauf der Güte und Bogenwinkelsumme der jeweils für eine Temperatur akzeptierten Rohrstrecke des Beispiels aus Abbildung 4.12 links oben. Die Zahl der Rohrbögen ist für die in der jeweiligen Temperatur synthetisierte, nicht für die jeweils akzeptierte Rohrstrecke dargestellt.

Die Tabelle 6 listet die Laufzeiten²⁴ der Verrohrungssprache des einsträngigen Beispiels unter Verwendung des Referenzpfades (Abbildung 4.12 links oben), nach Regelaufrufen aufgeschlüsselt, auf. Dabei wird für die Regeln *B-E*, welche einen Aufruf der KKF-Konstruktion beinhalten, zwischen den Regelaufrufen unterschieden, die innerhalb der KKF-Konstruktion auf eine neue Rohrstrecke führen und denen, die auf Grund der geometrischen Gegebenheiten keine mit den gegebenen Zwangsbedingungen kompatible Rohrstrecke finden können.

Die erfolglosen Regelaufrufe, ohne Auffinden einer neuen Rohrstrecke, sind in kleingedruckten Buchstaben aufgeführt. Im Falle eines erfolglosen Regelaufrufes wird die Bestimmung der Rohrstrecken mit der zuletzt akzeptierten Konfiguration fortgesetzt.

Die Analyse der Laufzeiten ergibt, dass über die Hälfte der Laufzeit in der KKF-Konstruktion verbracht wird (Regeln *B,b,C,c,D,d*). Von diesem Teil der Laufzeit führen über dreiviertel der Laufzeit, auf Grund der geometrischen Gegebenheiten, nicht auf eine neue Konfiguration.

Die Laufzeiten der Regeln E_1 und E_2 im Starrkörpersimulator müssen begrenzt und damit als Parameter vorgeben werden, da diese Regeln nicht in allen Fällen auf einen stationären

²⁴Die Untersuchungen wurden auf einem PC mit 2 GB Arbeitsspeicher und zwei Prozessorkernen mit jeweils 2 GHz durchgeführt.

Regel	Laufzeit	relative Laufzeit
Regel <i>A</i>	110 s	0.47 %
Regel <i>B</i>	1372 s	5.83 %
Regel <i>b</i>	9827 s	41.79 %
Regel <i>C</i>	406 s	1.73 %
Regel <i>c</i>	389 s	1.65 %
Regel <i>D</i>	275 s	1.17 %
Regel <i>d</i>	245 s	1.04 %
Regel <i>E</i> ₁	5390 s	22.92 %
Regel <i>E</i> ₂	5502 s	23.40 %
Gesamt	23516 s	100.0 %

Tabelle 6: Laufzeiten der Verrohrungssprache mit Optimierung auf Basis der Simulierten Abkühlung, aufgeschlüsselt nach Regeln und Status des KKF-Konstruktion Aufrufes: Regeln mit erfolglosem Aufruf der KKF-Konstruktion sind kleingedruckt aufgeführt.

Endzustand führen. So kann es durch die Kollisionsbehandlung, beim Aufbringen einer Kraft auf die Rohrstrecke, zu einer endlos oszillierenden "Ping-Pong-Bewegung" der Rohrstrecke kommen.

Die in den konkreten Beispielen gewählte Simulationsdauer beträgt $t_{E_x} = 5s$. Die Wahl der Simulationsdauer wird von den Bewegungsgeschwindigkeiten der Rohrstreckenelemente im Starrkörpersimulator bestimmt. Diese Geschwindigkeiten hängen von den Beträgen der aufgebrachten Kräfte F und der Massen m des Ersatzmodells der Rohrstrecke ab. Für kleine Massen und große Kräfte ergeben sich nach dem Newtonschen Gesetz,

$$a \propto F \cdot m, \quad (4.33)$$

große Werte für die Beschleunigung und somit, in der Annahme einer gegebenen Kraft, große Verschiebungen δx der Rohrstrecke in einem gegebenen Zeitintervall δt :

$$\delta x \propto a \cdot \delta t^2. \quad (4.34)$$

Damit sollten die aufgebrachten Kräfte, für das Ziel geringer Simulationsdauern, möglichst groß und die Massen des Ersatzmodells möglichst klein gewählt werden. Dies ist aber bezüglich der numerischen Stabilität, der das Ersatzmodell der Rohrstrecke beschreibenden Differentialgleichungen, nicht unkritisch [210, 205]. Typische Beträge der Kräfte und Massen wurden mit Testläufen bestimmt. Die für die Beispiele gewählten Parameter ($m = O(1kg)$, $F = O(100N)$, $t = O(1s)$) entsprechen einem numerisch robustem Setup.

Die am Beispiel des Würfels durchgeführten Untersuchungen zeigen, dass die implementierte Verrohrungssprache mit Simulierter Abkühlung erlaubt, mit den gegebenen Randbedingungen kompatible Rohrstrecken aus Standardrohrbögen zu erzeugen. Auf Grund des vereinfachten Abkühlungsplanes ist allerdings nicht zu erwarten, dass das globale Optimum der Verrohrung gefunden wird.

Im folgenden Abschnitt wird die Übertragbarkeit des in diesem Abschnittes dargelegten Konzeptes auf die Synthetisierung von Rohrstrecken in einem industriellen Anwendungsfall untersucht.

Anwendung: Verrohrung Frachtschiffskamin In diesem Abschnitt wird das zuvor definierte Regelwerk, basierend auf dem Optimierungsschema der Simulierten Abkühlung, auf das Beispiel der Verrohrung des Abgaskamines des Frachtschiffes aus Kapitel 3.2.2 angewandt. In diesem

Beispiel sind vier parallele SCR-Systeme im gegebenen Bauraum des Schiffes zu entwerfen. Die Positionen der Motoren und damit der Motorauslässe sowie die Lage der Mischstrecken und der SCR-Boxen werden von der entsprechenden Teilentwurfssprache übernommen.

In der nachfolgenden Darstellung sollen nur die Verrohrungen zwischen den Motoren und den Mischstrecken betrachtet werden, da diese, auf Grund der geometrischen Gegebenheiten, den höchsten Anspruch an die Verrohrungssprache stellen. Die Information über den zur Verfügung stehenden Bauraum soll wiederum über die Verwendung eines Referenzpfades berücksichtigt werden. Dazu wird für jeden Motor, entsprechend Kapitel 4.1.4, ein Pfadsuchalgorithmus auf dem Graphen Ω des diskretisierten Bauraumes ausgeführt. Die den Motorauslässen beziehungsweise der Eingänge der Mischstrecke jeweils nächstgelegene Zelle von Ω dienen dann in der Bestimmung des Referenzpfades als Start- beziehungsweise Endzellen des Pfadsuchalgorithmus.

Die Gewichtswerte $|e_k|$ der Kante $e_k \in E$ des Graphen Ω sollen in dem betrachteten Beispiel nur aus einem ortsabhängigen Parameter $f(\vec{x}_k)$ bestehen:

$$|e_k| = f(\vec{x}_k) = \frac{1}{\min(d(\vec{x}_k, \text{Bauraumgrenze}))}. \quad (4.35)$$

Die Kantengewichte in Gleichung 4.35 werden aus der Inversen des minimalen Abstandes zu der Bauraumgrenze, des der Kante zugeordneten Punktes \vec{x}_k aus Gleichung (4.17), bestimmt. Der Referenzpfad wird mit diesen Kantengewichten, unter Minimierung der entlang des Pfades gebildeten Gewichtssumme, bestimmt. Dieser kombiniert die Eigenschaften eines möglichst kurzen (Anzahl der Summanden) und eines wandfernen Weges (Wert der Summanden) zwischen den gegebenen Pfadenden.

Diese Wahl entspricht der physikalisch motivierten Forderung nach einem möglichst kurzen Pfad zur Minimierung des Gegendruckes, welcher auf Grund der hohen Temperaturen des Abgasrohres in möglichst großen Abständen von anderen Komponenten, damit auch weit weg von den Bauraumgrenzen, verlaufen soll.

Für die Synthese der Verrohrung auf Basis der so gefundenen Referenzpfade sollen Rohrbögen mit folgenden Winkeln $\bar{\alpha}$ zur Verfügung stehen: $\bar{\alpha} = \{15^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ\}$. Der Rohrdurchmesser beträgt $D = 0.5m$ und entspricht dem Auslassdurchmesser vom Motor. Die Biegeradien der Rohrbögen seien mit $R = D$ vorgegeben. Daraus lässt sich die Ausdehnung der Rohrbögen nach Gleichung (4.7) berechnen.

Es soll der vereinfachte Abkühlungsplan des vorherigen Prinzipbeispiels, der Verrohrung in einem Würfel, wiederverwendet werden. Die zu minimierende Gütefunktion der Simulierten Abkühlung soll sich aus drei Teilzielen zusammensetzen:

1. Minimierung des Abstandes zum Referenzpfad für die Einhaltung der Bauraumgrenzen und zur Minimierung der Verrohrungslänge.
2. Minimierung der Bogenwinkelsumme zur Reduktion des Gegendruckes durch die Vermeidung von (unnötigen) Strömungsumlenkungen.
3. Verhinderung gegenseitiger Durchdringung der Rohrstrecken.

Die Güte bezüglich des Abstands der Strecke I' zum Referenzpfad wird entsprechend Gleichung (4.26) des vorherigen Beispiels berechnet:

$$W_{Ref}(I') = d(\text{Rohrstrecke}, \text{Referenzpfad}). \quad (4.36)$$

Analog dazu findet die Berechnung der Güte von Rohrstrecke I' zur Bewertung der Bogenwinkelsumme entsprechend Gleichung (4.7) des vorherigen Beispiels statt:

$$W_\alpha(I') = \sum_k \bar{\alpha}_k. \quad (4.37)$$

Als letzte Bewertungsgröße soll wiederum die gegenseitige Durchdringung der Rohrstrecken eingeführt werden. Diese soll, wie im vorherigen Beispiel in den Gleichungen (4.32) und (4.28)), als Faktor in der Gütefunktion berücksichtigt werden. Damit ergibt sich folgende, zu minimierende, Gesamtgütefunktion für die Rohrstrecke I' . Die Abhängigkeit der Güte vom Zeitschritt t ist in diesem Beispiel nicht explizit dargestellt:

$$W(I') = W_\alpha(I') \cdot W_{Ref}(I') \cdot w(\bar{D}_t(I')). \quad (4.38)$$

Abbildung 4.14 zeigt das Ergebnis der Synthese des Rohrabschnittes vom Motor zu den Mischstrecken im Bauraum des Frachtschiffes unter Verwendung der Gütefunktion (4.38) und dem Abkühlungsschema des vorherigen Beispiels. Die im gezeigten Beispiel schon dargestellten Verrohrungen der weiteren Abschnitte des Abgassystems wurden ebenfalls mit der Verrohrungssprache generiert, werden aber im weiteren Verlauf nicht modifiziert.

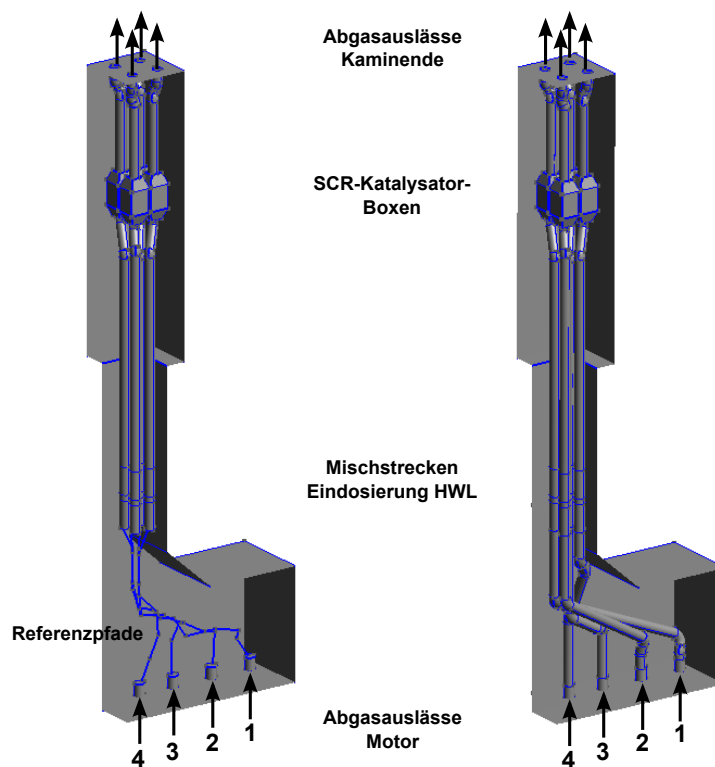


Abbildung 4.14: Abgaskamin des Frachtschiffes aus Kapitel 3.2.2 mit vier Motoren. Der Rohrabschnitt von Motorauslass bis Eintritt Mischstrecke wird mit der Verrohrungssprache mit Simulierter Abkühlung bestimmt. Linke Seite: Referenzpfad des zu verrohrenden Abschnittes. Rechte Seite: CAD-Modell der in der Verrohrungssprache synthetisierten Verrohrung. Die Indexierung der Strecken erfolgt nach den Motorauslässen von rechts nach links.

Die finalen Rohrbogenkonfigurationen der einzelnen Rohrstrecken sind in Tabelle 7 aufgeführt. Die Rohrbogenkonfiguration bei Ausführung des Axioms bestand, identisch für alle Rohrstrecken, aus acht Rohrbögen mit einem Bogenwinkel von jeweils $\bar{\alpha}_k = 45^\circ$. Die in Abbildung 4.14

auf der rechten Seite dargestellten, bezüglich der gegebenen Randbedingungen (keine gegenseitige Durchdringung und Verletzung der Bauraumgrenzen, Verwendung vorgegebener Rohrbogenwinkel) validen Rohrstrecken, zeigen exemplarisch die erfolgreiche Anwendung des Regelwerkes auf Basis der Simulierten Abkühlung.

Rohrstrecke	Bogenwinkel $\{\bar{\alpha}_1, \dots, \bar{\alpha}_k, \dots, \bar{\alpha}_{N-2}\}$	Bogenwinkelsumme
1	$\{30^\circ, 90^\circ, 30^\circ, 60^\circ\}$	210°
2	$\{15^\circ, 60^\circ, 60^\circ\}$	135°
3	$\{90^\circ, 45^\circ, 60^\circ\}$	195°
4	$\{30^\circ, 30^\circ, 45^\circ\}$	105°

Tabelle 7: Bogenwinkelkonfigurationen, der mit Simulierter Abkühlung synthetisierten Rohrstrecken des Abschnitts von Motor bis Eintritt Mischstrecke.

Die Gesamtlaufzeit der Synthese der Verrohrung in Abbildung 4.14 beträgt circa 30h. Die Synthese läuft komplett autonom ab und bedarf nach dem Start keiner weiteren Überwachung.

Abbildung 4.15 zeigt den Verlauf der Gütefunktion (Gleichung (4.38)) der einzelnen Rohrstrecken gegen den Temperaturparameter sowie den Verlauf der Summe der Gütefunktionen über die Rohrstrecken. Es ist zu erkennen, dass die Gesamtgüte der finalen Konfiguration dem globalen Minimum entspricht. Dies gilt aber nicht für die Gütefunktionen der einzelnen Rohrstrecken. Diese weisen, im Vergleich zur Güte der finalen Konfiguration, wesentlich geringere Gütewerte im Verlauf der Synthese auf. In dem gezeigten Beispiel führen die gleichzeitig ablaufenden Optimierungen der Einzelstrecken, welche nur über die Kollisionsdetektion des Starrkörpersimulators im Konfigurationsraum und über die Gütefunktion (4.28) der gegenseitigen Durchdringungen gekoppelt sind, auf eine Gesamtkonfiguration mit einer, bezüglich der synthetisierten Konfigurationen, minimalen Gütesumme.

Die prinzipielle Machbarkeit der Synthese von Rohrstrecken mit vorgegebenen Rohrbögen unter Verwendung der Simulierten Abkühlung als Optimierungswerkzeug konnte mit den beiden Beispielen (Prinzipdemonstration und Frachtschiffkamin) erfolgreich demonstriert werden. Es ergeben sich jedoch noch erhebliche Ansprüche an die Rechenleistung, um die bisher recht hohen Laufzeiten, die sich im Stundenbereich bewegen, zu reduzieren.

In der Zukunft kann, mit steigenden Rechenleistungen, die Behandlung der Bauraumgrenzen direkt in der Kollisionsbehandlung des Starrkörpersimulators durchgeführt werden. Somit kann eine Bauraumverletzung zuverlässig, ohne die Verwendung eines im bestehenden Bauraum berechneten Referenzpfades, ausgeschlossen werden. Eine Behandlung der Bauraumgrenzen im Starrkörpersimulator wurde im Laufe der Arbeit, mit Hilfe der vom Starrkörpersimulator ODE [177] zur Verfügung gestellten Funktionalitäten, implementiert. Die Laufzeiten für in der Praxis typischerweise vorkommende Bauräume liegt im Bereich von Tagen und wurde deshalb nicht weiter verfolgt.

Die durchgeführten Untersuchungen, unter Verwendung eines innerhalb des Bauraumes verlaufenden Referenzpfades zur Einhaltung der Bauraumgrenzen, weisen eine für den Einsatz im frühen Konzeptentwurf ausreichende Güte der generierten Verrohrungen auf. Etwaige Bauraumverletzungen können gegebenenfalls nach erfolgter Synthese der Strecken mit der Kollisionsbehandlung des Starrkörpersimulators detektiert werden. Der zusätzliche Laufzeitaufwand der einmaligen Kollisionsbehandlung der Rohrstrecken mit den Bauraumgrenzen ist im Vergleich zur Gesamtlaufzeit akzeptabel.

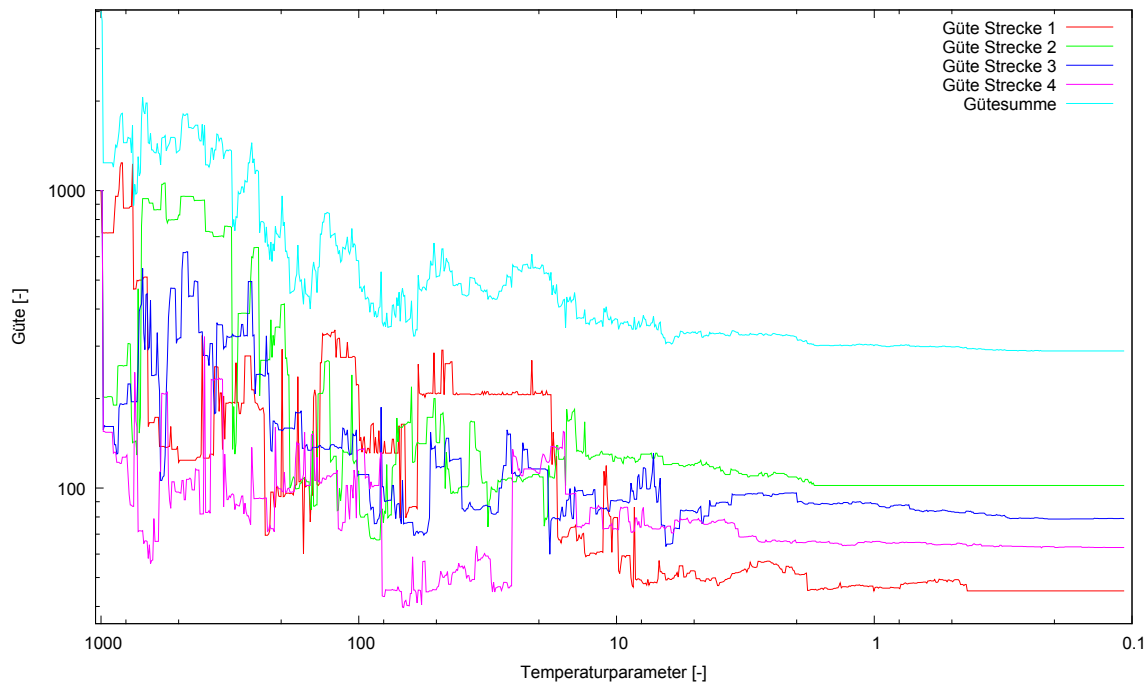


Abbildung 4.15: Verlauf der Güte der einzelnen Rohrstrecken sowie der Gesamtgüte für die Verrohrungssprache auf Basis der Simulierten Abkühlung am Beispiel der Synthese der Verrohrungen der SCR-Systeme des Frachtschiffes.

Gerade die Verwendung der KKF-Konstruktion macht eine regelbasierte Variation der Rohrbogenkonfiguration einer bestehenden Rohrstrecke möglich. Die Manipulation der Rohrstreckenkonfiguration mit Hilfe der KKF-Konstruktion stellt eine lokal, an einem betrachteten Rohrbogen inklusive der direkten Nachbarn, wirkende Regel dar. Durch den lokalen Charakter der Manipulation kann im Fortgang der Optimierung eine Rohrstrecke Stück für Stück manipuliert und an das Optimum geführt werden. Es bleibt aber zu untersuchen, ob beispielsweise der Einsatz eines anderen Optimierungsalgorithmus sowohl bezüglich der Ergebnissgüte als auch der Laufzeit vergleichbare oder gar bessere Ergebnisse liefert.

4.2.2. Parametrische Optimierung - Reinitialisierung

Im Vergleich zur Synthese von Rohrstrecken mit der Simulierten Abkühlung des vorherigen Abschnitts, verspricht die rein parametrische Optimierung einer vorgegebenen Rohrbogenkonfiguration, auf Basis des in Abschnitt 4.1.5 definierten Produktionssystems, eine schnellere Generierung von Rohrstrecken im Rahmen des Konzeptentwurfes.

Das Axiom (Regel A) aus Kapitel Kapitel 4.1.1 wird mit den gegebenen Zielkonfigurationen (Bogenwinkel, Biegeradien) sowie den jeweiligen Randbedingungen (Start-/Endpositionen und Richtungen) der Rohrstrecken aufgerufen. Im Axiom wird versucht, eine mit den Randbedingungen verträgliche initiale Rohrstreckenkonfiguration zu bestimmen.

Wie in der Darstellung des heuristischen Axioms in Kapitel 4.1.1 erläutert, kann es zu Situationen kommen, dass keine initiale Rohrstrecke bestimmt werden kann. Mit den Randbedingungen unverträgliche Rohrbogenkonfigurationen würden im Axiom zu einer endlosen Laufzeit führen. Aus diesem Grund sollte die Laufzeit des Axioms limitiert werden. Dies kann für zu gering gewählte Werte der zulässigen Laufzeit t_{Axi} aber dazu führen, dass theoretisch mit den Rand-

bedingungen verträgliche Rohrbogenkonfigurationen innerhalb der zulässigen Laufzeit nicht initialisiert werden können und so, obwohl theoretisch vorhanden, nicht gefunden werden.

Die parametrische Optimierung der im Axiom gewonnenen und mit den Randbedingungen kompatiblen initialen Rohrstrecken findet durch Aufruf der Regeln E_1 und E_2 (bei Verwendung eines Referenzpfades) entsprechend Abschnitt 4.2.1 statt. Die Laufzeiten der Regelaufrufe der parametrischen Optimierung wird dabei ebenso auf eine maximale Laufzeit t_{E_x} begrenzt. Damit stellt die Synthese der Rohrstrecken auf Basis der dargestellten parametrischen Reinitialisierung ein heuristisches Verfahren dar.

Prinzipbeispiel: Würfel Die Synthese von Rohrstrecken auf Basis des Produktionssystems mit Reinitialisierung der Rohrstrecken bei Modifikation der Rohrbogenkonfiguration soll in diesem Abschnitt dargestellt werden. Dabei wird wiederum auf den Bauraum des Würfels, die Verrohrungsrandbedingungen sowie den Referenzpfad von Abschnitt 4.2.1 zurückgegriffen. Dies umfasst die gleichzeitige Verrohrung dreier Rohrstrecken, diagonal durch einen Würfel, mit Überkreuzung der äußeren Strecken. Die Aus- und Einlässe der Strecken sind entlang einer Achse des Würfels angeordnet und in die selbe Richtung ausgerichtet (Abbildung 4.17 links).

Die zu untersuchenden Rohrbogenkonfigurationen sollen aus einer variablen Anzahl von Rohrbögen mit einem festen Rohrbogenwinkel von $\bar{\alpha} = 45^\circ$ bestehen. Die jeweiligen Konfigurationen werden durch Reinitialisierung der entsprechenden Rohrstrecken untersucht. Die parametrische Optimierung der Strecke soll mit Regel E_2 (Anziehung der Rohrstrecke an den Referenzfad), unter Verwendung eines frei gewählten und für die drei Rohrstrecken gemeinsamen Referenzpfades, untersucht werden. Die Laufzeit des Axioms sei auf maximal $t_{Axi} = 150s$ begrenzt und Regel E_2 soll für jeweils $t_{E_x} = 120s$ ausgeführt werden. Diese Werte stellen einen Kompromiss zwischen Laufzeit und dem Auffinden einer stabilen Zielkonfiguration dar, welche auf den Ergebnissen empirischer Voruntersuchungen basieren.

Abbildung 4.16 zeigt beispielhaft die Güte der so gewonnen Strecken für verschiedene Rohrbogenkonfigurationen unter Verwendung der Gütefunktion (4.26), welche den dimensionslosen Abstand zum Referenzpfad repräsentiert. Aufgetragen sind verschiedene Anzahlen von Rohrbögen von zwei der Strecken bei festgehaltener Rohrbogenzahl der dritten Strecke.

Es ist zu erkennen, dass für eine größere Zahl von Rohrbögen ein geringerer Abstand zu dem Referenzpfad, auf Grund der höheren Anzahl an Freiheitsgraden der Rohrstrecken, realisiert werden kann. Es finden sich jedoch auch Rohrbogenkonfigurationen (grau dargestellt) für die im Axiom in der gegebenen Laufzeit keine mit den Randbedingungen verträglichen Streckenführungen, z.B. auf Grund gegenseitiger Blockade der Rohrstrecken, gefunden werden konnten.

Abbildung 4.17 stellt die CAD-Modelle zweier mit der parametrischen Optimierung bestimmten Verrohrungen des Würfels dar. Es sind die Rohrstrecken mit den Rohrbogenkonfigurationen aus Abbildung 4.16 links unten (jede der drei Strecken besteht aus vier Rohrbögen) und der Konfiguration rechts oben von Abbildung 4.16 aufgetragen (eine Strecke mit vier Bögen und jeweils zehn Bögen in den anderen beiden Strecken).

Das in dem Prinzipbeispiel gezeigte "durchrastern" verschiedener Rohrbogenkonfigurationen kann im Vorentwurf von SCR-Systemen beispielsweise zur Abschätzung der zielführenden Bogenkonfiguration eingesetzt werden. Dazu können verschiedene Anzahlen von Rohrsegmenten mit vergleichsweise kleinen Bogenwinkeln untersucht und gegen den Abstand zum Referenzpfad aufgetragen werden.

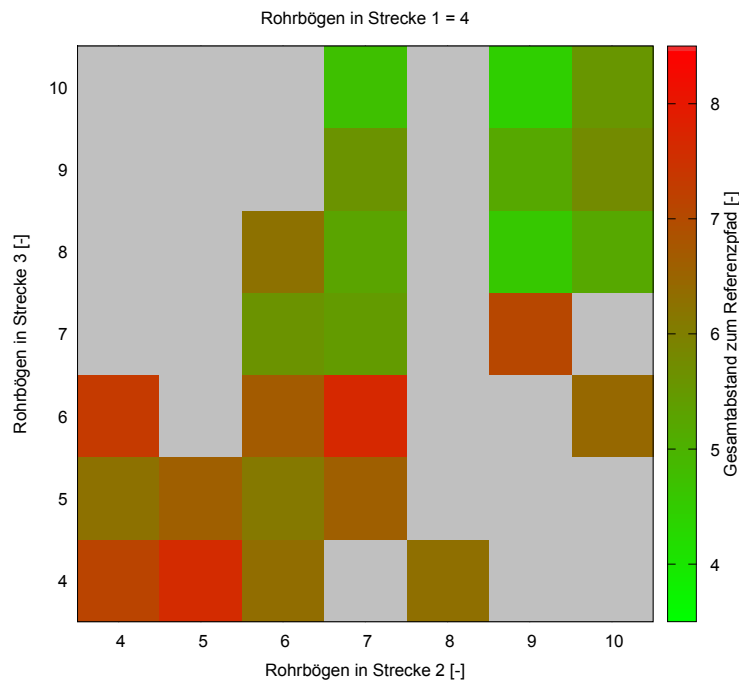


Abbildung 4.16: Über die Rohrstrecken summierter, dimensionsloser Abstand zum Referenzpfad (Gütefunktion (4.26)) verschiedener Rohrbogenkonfigurationen (vier Rohrbögen in Rohrstrecke 1) am Beispiel der Verrohrung des Würfels aus Abbildung, 4.17 bestimmt durch Reinitialisierung der Rohrstrecken und parametrischer Optimierung.

Bei Verwendung solcher Rohrelemente kann dann aus der Anzahl der Rohrbögen heuristisch auf Rohrbogenkonfigurationen aus Elementen größerer Bogenwinkel²⁵ geschlossen werden: Im Produktionssystem könnte eine Auswahlregel modelliert werden, die nahe beieinander liegende Rohrsegmente der Strecke zu einem Standardrohnbogen mit größerem Bogenwinkel zusammenfasst. Ein solches Regelwerk könnte damit auf eine zielführende Kombination von Standardrohnbögen führen, ohne das zeitaufwändigere Optimierungsverfahren der Simulierten Abkühlung anwenden zu müssen.

Anwendung: Verrohrung Frachtschiffskamin In diesem Abschnitt wird der Entwurf von Rohrstrecken mit Hilfe des Prinzips der Reinitialisierung der Rohrstrecken, wiederum am Beispiel des Abgaskamines des Frachtschiffes aus Kapitel 3.2.2, gezeigt. Dies umfasst, entsprechend der in der Simulierten Abkühlung dargestellten Verrohrung des Abgaskamines, den gleichzeitigen Entwurf der Rohrstrecken für vier Motoren im gegebenen Bauraum. Im Gegensatz zur Simulierten Abkühlung und entsprechend dem vorherigen Beispiel soll die Rohrbogenkonfiguration nun manuell vorgegeben werden (Versuch und Irrtum).

Es sollen drei Fälle für den Verrohrungsabschnitt von Motor zum Eintritt der Mischstrecken betrachtet werden:

- **Rechtwinklig:** Generierung der Verrohrung mit Rohrbögen von $\bar{\alpha} = 90^\circ$.
- **Segmentiert:** Generierung der Verrohrung mit Rohrbögen von $\bar{\alpha} = 20^\circ$.
- **Ausgewogen:** Generierung der Verrohrung mit Rohrbögen von $\bar{\alpha} = 30^\circ$ und $\bar{\alpha} = 60^\circ$.

²⁵Typische Werte in der Praxis sind: 30° , 45° , 60° , 90°

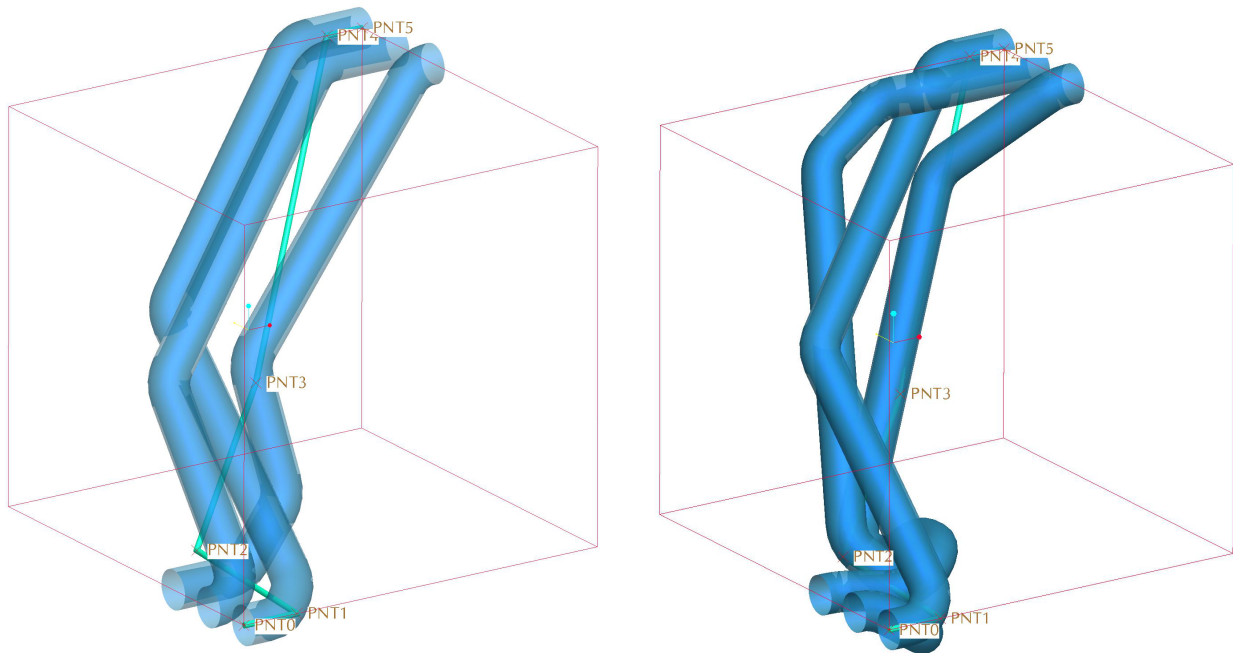


Abbildung 4.17: Prinzipdemonstration der Verrohrungssprache mit Reinitialisierung der Rohr-
strecken jeweils unter Verwendung der Gütefunktion (4.26). Linke Seite: Rohr-
bogenkonfiguration aus Abbildung 4.16 links unten. Rechte Seite: Rohr-
bogenkonfiguration aus Abbildung 4.16 rechts oben.

Die ersten beiden Konfigurationen sollen stellvertretend für folgende Extremfälle stehen: Die Verwendung von rechtwinkligen Bögen ist eine in der Verfahrenstechnik und dem Anlagenbau verbreitete Entwurfsregel [163]. Die Verwendung von Rohrsegmenten mit kleinen Bogenwinkeln stellt eine Annäherung an den Grenzfall einer frei gebogenen Verrohrung dar. Diese Rohrbogenkonfiguration wird voraussichtlich aus der anzahlmäßig größten Zahl an Rohrbogenelementen bestehen. Die dritte Rohrbogenkonfiguration soll stellvertretend für eine ausgewogene Verrohrung stehen, die keine rechtwinkligen Verläufe erzwingt, gleichzeitig aber eine geringe Anzahl von Bogenelementen umfassen soll.

Die Verrohrungen der anderen Abschnitte, von der Mischstrecke zu den SCR-Katalysatorboxen und von den SCR-Boxen zum Kaminende, sollen wiederum auf Grund des geringeren Schwierigkeitsgrades dieser Verrohrungsabschnitte nicht explizit dargestellt werden. Die Verrohrungen dieser Abschnitte werden aber, wie im Beispiel der Simulierten Abkühlung, ebenfalls unter Verwendung der Verrohrungssprache erzeugt. Die Abbildung 4.18 zeigt neben den Referenzpfaden, welche identisch zu den Referenzpfaden des Beispiels des Frachtschiffes aus Abschnitt 4.2.1 sind, die durch Reinitialisierung bestimmten Rohrstrecken für die drei Rohrbogenkonfigurationen.

Die Tabelle 8 zeigt die Bogenwinkel der in Abbildung 4.18 dargestellten Ergebnisse inklusive der jeweiligen Bogenwinkelsumme für die drei betrachteten Rohrbogenkonfigurationen. In der manuellen Ausführung der Synthese der Rohrstrecken, nach Versuch und Irrtum, erwies es sich

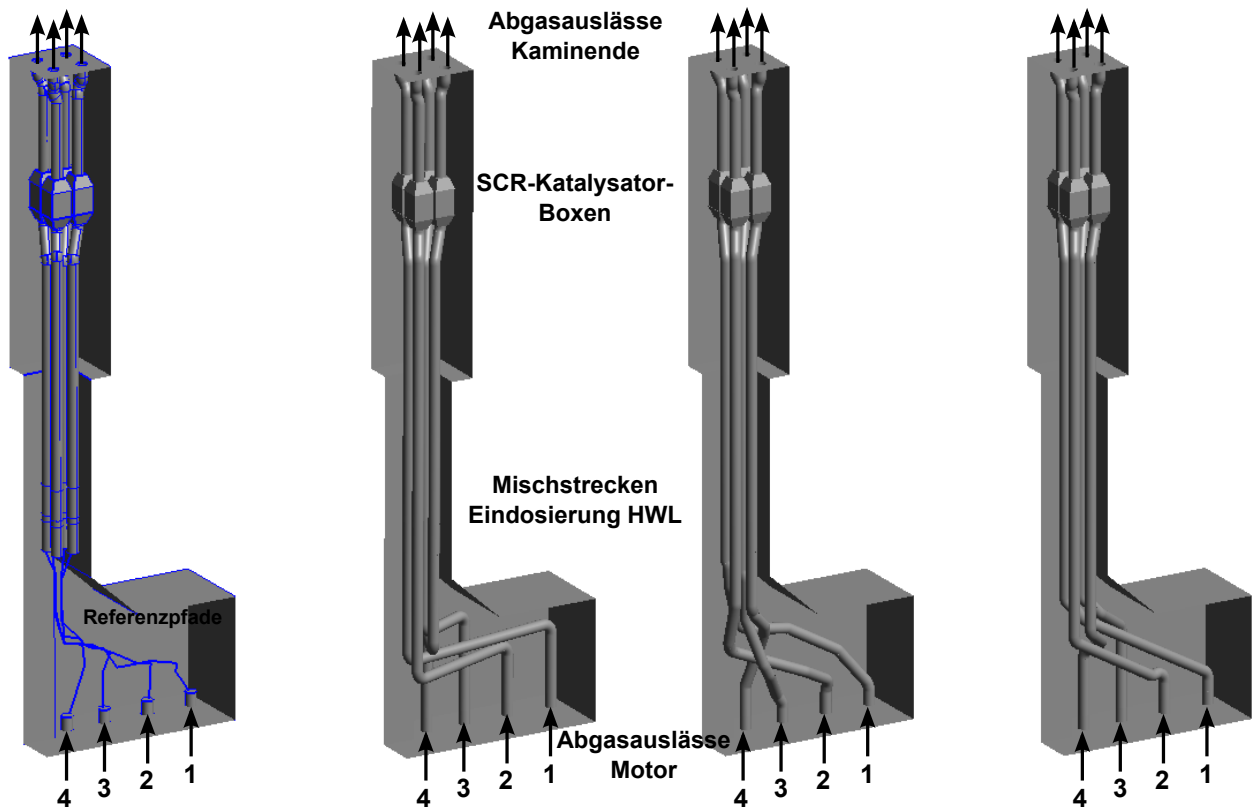


Abbildung 4.18: Abgaskamin des Frachtschiffes aus Kapitel 3.2.2 mit vier Motoren. Der Rohrabschnitt von Motorauslass bis Eintritt Mischstrecke wird mit der Verrohrungssprache mit manuell vorgegebenen Rohrbogenkonfigurationen mittels Reinitialisierung bestimmt (Versuch und Irrtum). Maximale Laufzeiten der Regeln: $t_{Ax_i} = 120s$ und $t_{E_2} = 60s$. Von links nach rechts: Referenzpfad entsprechend Abbildung 4.14, rechtwinklige Rohrbögen, Rohrbögen mit $\bar{\alpha} = 20^\circ$, ausgewogene Rohrbogenkonfiguration ($\bar{\alpha} = 30^\circ$ und $\bar{\alpha} = 60^\circ$).

als schwierig, Rohrstreckenkonfigurationen mit ähnlich niedrigen Gesamtbogenwinkelsummen zu synthetisieren, wie sie mit Hilfe des automatisierten Optimierungsschemas der Simulierten Abkühlung realisiert werden konnten. Dies ist beim Vergleich der Bogenwinkelsummen der beiden Synthesevarianten in den Tabellen 7 und 8 zu erkennen.

Die Laufzeit für die manuelle Synthese der Rohrstrecken liegt im Bereich weniger Minuten für eine gegebene Rohrbogenkonfiguration. Für die Durchführung dieser Synthese wurde eine graphische Benutzeroberfläche implementiert, die einen manuellen Aufruf der einzelnen Regeln, mit manuell vorgegebenen Parametern, erlaubt. Die Regelausführung kann dann in der graphischen Oberfläche visuell überwacht werden.

4.2.3. Lösungsmenge der Verrohrung

Abschließend soll in diesem Abschnitt die Lösungsmenge der mit der Verrohrungssprache generierbaren Verrohrungen betrachtet werden. Dazu soll zunächst von einem festen Startpunkt $P_S = \vec{P}_0$ der Verrohrung und der Startrichtung $\vec{D}_S = \vec{D}_0$ ausgegangen werden. Für die folgende Darstellung soll der einfache Fall einer Verrohrung, bestehend aus zwei geraden Rohrstücken mit variablen Längen und einem dazwischen liegenden Rohrbogen, betrachtet werden. Wie in den

Rohrbogenkonfiguration		Bogenwinkel $\{\bar{\alpha}_1, \dots, \bar{\alpha}_k, \dots, \bar{\alpha}_{N-2}\}$	Bogenwinkelsumme
rechtwinklig	Strecke 1	$\{90^\circ, 90^\circ\}$	180°
	Strecke 2	$\{90^\circ, 90^\circ\}$	180°
	Strecke 3	$\{90^\circ, 90^\circ\}$	180°
	Strecke 4	$\{90^\circ, 90^\circ\}$	180°
segmentiert	Strecke 1	$\{20^\circ, \dots, 20^\circ\}$	200°
	Strecke 2	$\{20^\circ, \dots, 20^\circ\}$	200°
	Strecke 3	$\{20^\circ, \dots, 20^\circ\}$	160°
	Strecke 4	$\{20^\circ, \dots, 20^\circ\}$	160°
ausgewogen	Strecke 1	$\{60^\circ, 30^\circ, 30^\circ, 60^\circ\}$	180°
	Strecke 2	$\{60^\circ, 30^\circ, 30^\circ, 60^\circ\}$	180°
	Strecke 3	$\{60^\circ, 30^\circ, 30^\circ, 60^\circ\}$	180°
	Strecke 4	$\{60^\circ, 30^\circ, 30^\circ, 60^\circ\}$	180°

Tabelle 8: Bogenwinkelkonfigurationen der durch manuelle Reinitialisierung (Versuch und Irrtum) synthetisierten Rohrstrecken im Bauraum des Frachtschiffes (Abschnitt: Motor bis Mischstrecke) für die Rohrbogenkonfigurationen *rechtwinklig*, *segmentiert* und *ausgewogen*. Indexierung wie in Abbildung 4.18.

Ausführungen zur KKF-Konstruktion in Kapitel 4.1.3 gezeigt, liegen die möglichen Endpunkte eines Rohrbogens auf einem Kegelmantel.

Der Scheitelpunkt des Kegels entspricht dem Polygonpunkt des zugeordneten Polygonzuges (siehe Abbildung 4.3). Der Kegelmantel kann, entsprechend der zugelassenen variablen Länge der geraden Rohrstücke, entlang der Startrichtung verschoben werden. Der Platzbedarf des Rohrbogens ist zwar prinzipiell auch hier zu berücksichtigen (Gleichungen (4.8) bis (4.8)), soll aber in dieser Prinzipdarstellung der Lösungsmenge weggelassen werden.

Abbildung 4.19 deutet die Menge der Zielpunkte an, welche mit den dargestellten Verrohrungselementen (gerades Rohr - Rohrbogen - gerades Rohr), ausgehend vom gegebenen Startpunkt, erreicht werden können. Die Endrichtung eines auf dem Kegelmantel liegenden Endpunktes wird durch den Vektor, der vom Scheitelpunkt des Mantels zum Endpunkt zeigt, gebildet.

Die Menge der Zielpunkte P_E und Zielrichtungen \vec{D}_E einer von einem Startpunkt P_S , mit vorgegebener Startrichtung \vec{D}_S , ausgehenden Verrohrung, unter Verwendung einer gegebenen Rohrbogenkonfiguration I , stellt eine Teilmenge im sechsdimensionalen mathematischen Raum²⁶ dar, welcher von den Einheitsvektoren der Endpunkte und Endrichtungen aufgespannt wird. Diese Menge soll im Folgenden Zielmenge $Z(P_S, \vec{D}_S, I)$ der Verrohrung genannt werden.

Für eine Rohrbogenkonfiguration I können sich im Allgemeinen verschiedene Rohrstrecken mit identischen Endpunkten und Endrichtungen ergeben. Diese sollen die Lösungsmenge $L(P_S, \vec{D}_S, P_E, \vec{D}_E, I)$ der Rohrstrecken, zwischen vorgegebenen Start-/Endpunkten und Richtungen, für eine Rohrbogenkonfiguration I bilden. Ein entsprechendes Beispiel zeigt die Abbildung 4.11 der KKF-Konstruktion, welche für vorgegebene Start-/Endpunkte beziehungsweise Richtungen inklusive vorgegebener Rohrbögen zwei potentielle Zwischenpunkte und damit zwei potentielle Rohrstrecken bestimmen kann.

Die parametrische Variation der Rohrstrecke mit den Regeln E_x der dargestellten Verrohrungssprache stellt damit ein Durchlaufen der Lösungsmenge L für eine vorgegebene Kombination aus Start-/Endpunkt und Start-/Endrichtung dar, welche im Axiom (Regel A) für eine gegebene

²⁶Entspricht dem mathematischen Entwurfsraum nach Abbildung 6.2 in Abschnitt 6.1.2.

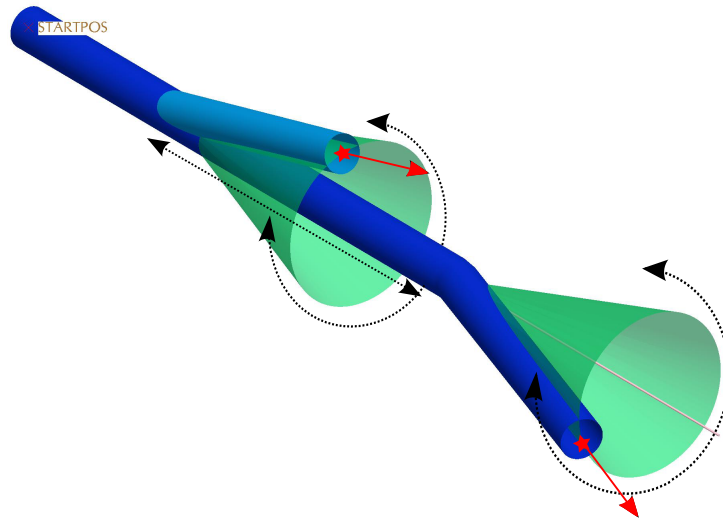


Abbildung 4.19: Freiheitsgrade einer Verrohrung mit den Rohrelementen: Gerade, Bogen, Gerade. Verschiebungsfreiheitsgrade der kegelförmigen Zielmenge (grün) sind mit gestrichelten Pfeilen angedeutet. Zwei mögliche Endpunkte inklusive Endrichtungen sind rot dargestellt.

ne Rohrbogenkonfiguration I gefunden wurde. Die Kombination aus Endpunkt und Richtung für eine vorgegebene Startposition und Richtung repräsentiert einen Punkt der Zielmenge Z .

Bei Betrachtung der Zielmenge, am Beispiel eines Rohrbogens (Rohrbogenkonfiguration I vorgegeben) in Abbildung 4.19, wird deutlich, dass zwar eine sehr große Menge (entlang der Startrichtung extrudierter Zylindermantel) an Endpunkten erreicht werden kann, die Wahl eines solchen Endpunktes aber im gegebenen Fall die Fixierung *eines* festen Wertes der Endrichtung bedeutet. Diese wechselseitige Einschränkung der Freiheitsgrade der Endpunkte und Endrichtungen kommt vor allem bei der Verrohrung weniger Rohrbögen zum Tragen.

Darüber hinaus ist am Beispiel der Verrohrungssprache zu erkennen, dass analog zur Mathematik [32] auch im Entwurf zwischen kontinuierlichen (Entwurfs-)Variablen und diskreten (Entwurfs-)Variablen unterschieden werden kann [3]. Im Beispiel der Verrohrungssprache stellt die Rohrbogenkonfiguration eine, in ihrer Kardinalität variable, endliche Menge diskreter Variablen dar: Die Bogenwinkel können nur die diskreten Werte der Winkel aus der Menge der zugelassenen Bogenwinkel annehmen. Dagegen stellen die Verschiebungsfreiheitsgrade der Verrohrung in Abbildung 4.19 kontinuierliche Variablen dar. Diese können auf Wertebereiche beschränkt sein, sind aber innerhalb der zugelassenen Grenze kontinuierlich.

In Form der mathematischen Beschreibung der beteiligten Domänen werden implizit diskrete und kontinuierliche Variablen in die digitale Entwurfsbeschreibung eingeführt. Wie in Kapitel 2.2.3 angedeutet, ist es ein typisches Merkmal der Standardisierung prinzipiell kontinuierliche Größen, wie z.B. Rohrdurchmesser, in diskrete Größen zu überführen, die z.B. in Form von Tabellenbüchern gestufte Rohrdurchmesser zum Standard erheben [104]. Analoge Standardisierungsnormen existieren für eine Vielzahl von Maschinen- und Konstruktionselementen.

Neben dieser Einschränkung der Definitionsmenge der Parametrischen Beschreibung von einer kontinuierlichen auf eine diskrete Menge, stellt die Topologie des den Entwurf repräsentierenden Entwurfsgraphen ein diskretes Konstrukt dar, welches explizit durch diskrete Knoten- und Kantenmengen repräsentiert werden kann.

Im Zusammenspiel der diskreten Rohrbogenkonfiguration mit den kontinuierlichen Größen der Verrohrung ergeben sich im Beispiel der Verrohrungssprache, letztendlich aus den geometrischen Zwangsbedingungen, Beschränkungen der Freiheitsgrade beziehungsweise der Zielmenge einer Verrohrung. Dieses Phänomen tritt auch im Entwurf anderer Objekte auf, wenn über das Zusammenspiel von Rand- und Zwangsbedingungen sowie den Anforderungen eine Lösungsmenge Schritt für Schritt eingegrenzt wird. Diese auf die sinnvollen Entwürfe eingeschränkten Lösungsmengen bilden Teilmengen im einbettenden mathematischen Entwurfsraum (vergleiche dazu auch das Kapitel 2.2.6 sowie Abbildung 6.2 in Kapitel 6.1.2).

Für diesen Abschnitt kann abschließend festgehalten werden, dass am Beispiel der Generierung von Verrohrungen die Existenz verschiedener Variablentypen demonstriert werden kann, welche unterschiedliche mathematische Eigenschaften besitzen (kontinuierlich gegenüber diskret). Darüber hinaus kommt es im Entwurf durch die Auswahl einer im Entwurf zu integrierenden Technologie und/oder Topologie zu einem spezifischen Zusammenspiel der gegebenen Randbedingungen mit den beschreibenden Entwurfsvariablen. Im Beispiel führt die einsträngige Verrohrung von Komponenten unter Verwendung von Standardrohrbauteilen auf einen spezifischen Satz von Entwurfsvariablen, die einer spezifischen Art und Weise miteinander in Wechselwirkung stehen. Die Werte der Entwurfsvariablen, welche gültige, mit den Rand- und Zwangsbedingungen kompatible, Entwürfe repräsentieren bilden Teilmengen im einbettenden mathematischen Raum.

Aus diesen Ergebnissen lassen sich im weiteren Verlauf der Arbeit Aussagen über die Reihenfolge der einzelnen Entwurfsschritte im Entwurfsprozess ableiten. Solcherlei Einflüsse der Freiheitsgrade der am Entwurf beteiligten Größen und Variablen auf den Entwurfsablauf und deren Verallgemeinerung auf andere Entwurfsobjekte werden in Kapitel 6 dargestellt und diskutiert.

Teil III.

Ordnungsmechanismen im wissensbasierten Entwurf

5. Selbstassemblierende Lösungspfadfindung

Wie in Kapitel 2.3.6 dargestellt, ist der Einsatz eines Lösungspfadgenerators zur automatisierten Lösung von Gleichungssystemen im Konzeptentwurf als Instrument der deklarativen Wissensverarbeitung unersetzlich. Damit können Systeme von Zwangsbedingungen oder analytischen Gleichungen deklarativ spezifiziert und daraus automatisiert eine prozedurale Lösungssequenz bestimmt werden. Ein Lösungspfadgenerator kann ein Netzwerk von Abhängigkeiten, in dem keine Lösungsreihenfolge definiert ist, in eine zeitliche Lösungssequenz überführen [155].

5.1. Lösungsweg und Methodik

In diesem Kapitel soll zunächst das Regelwerk des selbstassemblierenden Lösungspfadgenerators definiert und dargestellt werden. Die Regeln sollen dabei mit Hilfe von Symmetrieargumenten motiviert und begründet werden.

5.1.1. Lösungssequenz aus Abhängigkeiten

Die Abhängigkeit der Systeme bzw. Variablen x_1 und x_2 über $f(x_1, x_2)$ enthält zunächst keine Information darüber, in welcher gerichteten Abhängigkeit die beiden Systeme stehen. Erst durch die Deklaration eines gegebenen, beziehungsweise bekannten Systems (bekannter Variablen) kann mit Hilfe des Lösungspfadgenerators eine gerichtete Kausalkette zur Abarbeitung der Systemabhängigkeiten bestimmt werden [171]. Dabei soll eine Abhängigkeit $f_i(x_j)$, die nur ein System umfasst, so interpretiert werden, dass das entsprechende System (bzw. Variable) x_j als bekannt gilt und die entsprechende Abhängigkeit $f_i(x_j)$ keine weitere Berücksichtigung findet.

Im Weiteren wird, auf Grund der einfacheren Beschreibung, nur noch von Gleichungssystemen, Gleichungen und Variablen, anstatt von Abhängigkeiten und Systemen, gesprochen. Die Ergebnisse sind aber nicht auf Gleichungssysteme beschränkt, sondern treffen auch auf die Lösung kausaler Abhängigkeitsbeziehungen in und zwischen Systemen zu. Die konkrete Lösungspfadfindung soll anhand des folgenden Beispielsystems näher betrachtet werden:

$$\begin{aligned}
 f_1 &= f_1(x_1, x'_5) \\
 f_2 &= f_2(x_1, x_2, x'_6) \\
 f_3 &= f_3(x_2, x_3, x_4) \\
 f_4 &= f_4(x_2, x_4, x'_5, x'_6)
 \end{aligned}
 \tag{5.1}$$

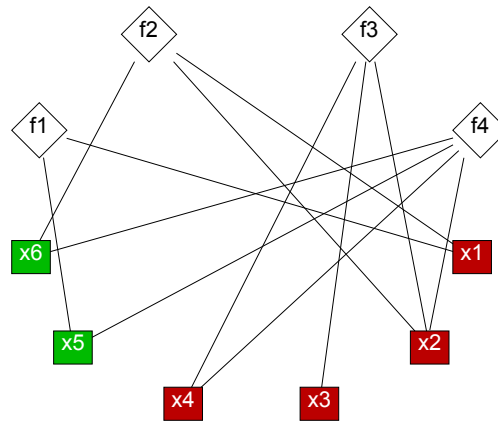


Abbildung 5.1: Bipartiter Graph des Beispielsystems (5.1).

Die Variablen x'_5 und x'_6 seien gegeben. In Abbildung 5.1 ist das System als bipartiter Graph dargestellt. Die Lösungssequenz dieses Systems kann mit dem graphenalgorithmischen Lösungspfadgenerator bestimmt werden:

$$\begin{aligned}
 x_1 & \text{ bestimmt durch } f_1 \\
 x_2 & \text{ bestimmt durch } f_2 \\
 x_4 & \text{ bestimmt durch } f_4 \\
 x_3 & \text{ bestimmt durch } f_3
 \end{aligned}
 \tag{5.2}$$

5.1.2. Lokale Symmetrie

Die Lösungssequenz 5.2 ist in Abbildung 5.2 direkt im, nun mit gerichteten Abhängigkeiten versehenen, bipartiten Graphen aufgetragen. Im Gegensatz zu der Darstellung in 2.3.6 wurde aber auf den Zwischenschritt der Eliminierung der Gleichungen, beim Übergang vom gematchten auf den gerichteten Graphen, verzichtet. Die Lösungssequenz erschließt sich wie in Abschnitt 2.3.6 prinzipiell aus der inversen topologischen Sortierung der Abhängigkeiten.

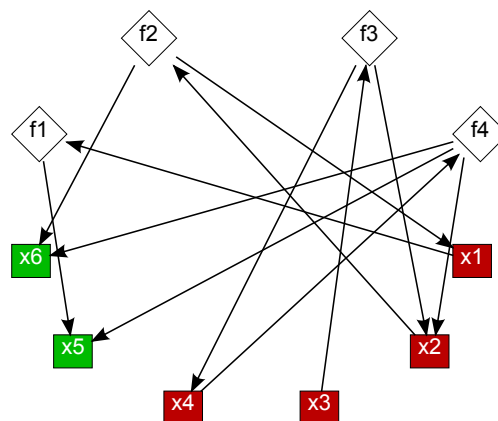


Abbildung 5.2: Bipartiter Graph des Beispielsystems (5.1) mit gerichteten Abhängigkeiten der Lösungssequenz. Gegebene Größen sind grün dargestellt.

Jeder Gleichungsknoten des ungerichteten Constraints Graph (z.B. 5.1) weist bei lokaler Betrachtung eine Symmetrie bezüglich der Vertauschung (Permutation) der verbundenen Va-

riablenknoten auf: Die ungerichteten Kanten können durch bidirektionale gerichtete Kanten äquivalent ersetzt werden. Bei Vertauschung der Nummerierung der mit dem betrachteten Gleichungsknoten verbundenen Variablenknoten ergibt sich keine Änderung des betrachteten Knotens und seiner aus- und eingehenden Kanten. Dies lässt sich besonders gut erkennen, wenn man die Kanten eines Knotens gleichwinklig anordnet (siehe Abbildung 5.3 linke Seite) und sich die lokale Symmetrie im ungerichteten Graphen als Rotationssymmetrie manifestiert.

Diese lokale Rotationssymmetrie (im Beispiel Rotationssymmetrie bei Drehung um $n \cdot 120^\circ$) der jeweiligen Gleichung und ihrer Kanten, welche die Abhängigkeiten widerspiegeln, wird durch die Einführung gerichteter Abhängigkeiten gebrochen: Die bidirektionalen, gerichteten Kanten werden jeweils durch unidirektionale Kanten ersetzt. Damit ändert sich die Symmetrie des betrachteten Gleichungsknotens bei Vertauschung der Nummerierung der verbundenen Variablenknoten. Dies ist wiederum in der gleichwinkligen Anordnung der Kanten des betrachteten Knotens zu erkennen (siehe Abbildung 5.3 rechte Seite).

Eine analoge Situation stellt sich bei den Variablen dar. Hier findet ein entsprechender Bruch der Symmetrie durch die Einführung der gerichteten Abhängigkeiten statt (Abbildung 5.4).

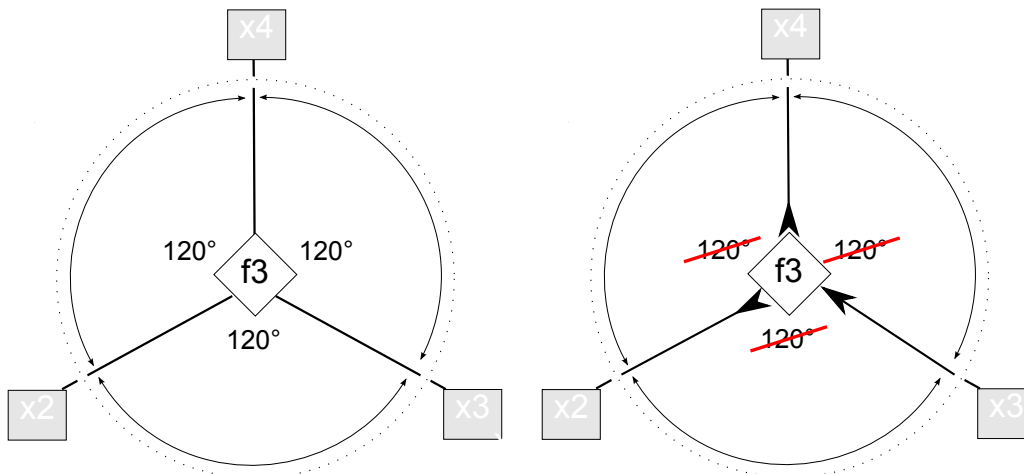


Abbildung 5.3: Lokale (innerhalb des gepunkteten Kreises) Rotationssymmetrie (links) der Gleichung f_3 mit Rotationswinkel von 120° wird durch Einführung gerichteter Abhängigkeiten (rechts) gebrochen.

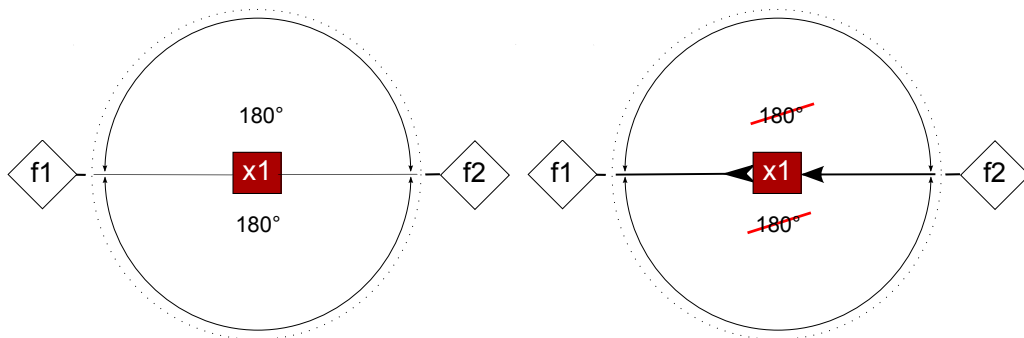


Abbildung 5.4: Lokale (innerhalb des gepunkteten Kreises) Rotationssymmetrie (links) der Variablen x_1 mit Rotationswinkel von 180° wird durch Einführung gerichteter Abhängigkeiten (rechts) gebrochen.

Dieser Symmetriebruch kann in Analogie zu physikalischen Systemen und deren Symmetrien gesehen werden: Nach Curie [46] ist für das Auftreten von Phänomenen (z.B. die Entstehung von Strukturen in Kristallgittern) in physikalischen Medien eine Reduktion der ursprünglichen Symmetrie des Mediums notwendig. In diesem Sinne wird ein Phänomen durch einen Symmetriebruch (Reduktion der Symmetrie) erst hervorgerufen [38]. Auf den Lösungspfadgenerator übertragen bedeutet dies, dass das Auffinden einer gerichteten Lösungssequenz im Abhängigkeitsgraphen und die Einführung einer Reihenfolge entlang einer Zeitachse direkt mit einem Symmetriebruch verknüpft ist.

5.1.3. Definition eines selbstassemblierenden, lokalen Regelwerkes

Vergleicht man die jeweiligen Muster aus den Abbildungen 5.3 und 5.4 mit dem, mit gerichteten Abhängigkeiten versehenen, bipartiten (Lösungs-)Graphen in Abbildung 5.2, so ergeben sich zwei verallgemeinerte Muster:

1. **Gleichungs-Knoten:**

Genau eine Kante ist von einer Variablen auf die Gleichung gerichtet. Die restlichen Kanten der Gleichung zeigen, von der Gleichung weg, auf andere Variablen.

2. **Variablen-Knoten:**

Genau eine Kante ist auf eine Gleichung gerichtet. Die restlichen Kanten der Variable zeigen, von anderen Gleichungen weg, auf die Variable selbst.

Dies wird bei der Betrachtung von Variable x_2 und Gleichung f_3 des Gleichungssystems (5.1) in Abbildung 5.2 besonders deutlich. Diese Muster spiegeln dabei folgende (mathematischen) Tatsachen wieder:

1. **Gleichung:**

Jede Gleichung kann zur Bestimmung genau einer enthaltenen Variable verwendet werden, wenn die übrigen in der Gleichung enthaltenen Variablen bekannt sind.

2. **Variable:**

Eine Variable wird durch genau eine Gleichung bestimmt und kann dann in weiteren Gleichungen als gegebene beziehungsweise bereits bestimmte Variable verwendet werden.

Daraus lassen sich die Regeln eines zeitdiskreten, dynamischen Systems definieren, im Folgenden *selbstassemblierender Lösungspfadgenerator* genannt. Dieses dynamische System soll, alternativ zu der sequenziellen Ausführung der in Kapitel 2.3.6 dargestellten einzelnen Graphenalgorithmen (Top-Down-Prinzip), zum Auffinden einer Lösungssequenz des Gleichungssystems verwendet werden können.

Es stellt damit die Realisierung eines Bottom-Up-Prinzips zur Suche des Lösungspfad dar. Dabei wird die Systemdynamik auf der Ebene der einzelnen Systemkomponenten (Variablen und Gleichungen) lokal definiert. Der Lösungspfad ergibt sich dann, als Struktur im Sinne der Definition aus Kapitel 2.5.4, selbstassemblierend aus dem Zusammenspiel der Einzelkomponenten auf unterster Ebene.

Ausgangspunkt des dynamischen Systems soll dabei der bipartite Graph der ungerichteten Abhängigkeiten der Gleichungen F und Variablen X sein, der direkt aus dem Gleichungssystem abgeleitet werden kann (siehe Abbildung 5.1 bzw. Abbildung 2.9 in Kapitel 2.3.6). Das Ziel des selbstassemblierenden Lösungspfadgenerators soll das Auffinden einer Lösungssequenz des Gleichungssystems sein.

Basierend auf den oben identifizierten Mustern können folgende Regeln definiert werden:

1. Transformationsregel Gleichungen:

Sind bis auf eine Variable alle Variablen einer Gleichung bekannt, so wird diese Variable durch die jeweilige Gleichung bestimmt.

2. Transformationsregel Variablen:

Ist eine Variable durch eine Gleichung bestimmt, so gelte sie auch in den anderen Gleichungen, in denen sie vorkommt, als bestimmte Variable.

Das dynamische System soll direkt auf der topologischen Struktur des bipartiten Graphen $G(X \cup F, E)$ der ungerichteten Abhängigkeiten definiert werden. Dazu wird jedem Variablenknoten $x_i \in X$ des Gleichungssystems eine boolesche Variable $\bar{x}_i \in \{0, 1\}$ zugeordnet. Diese booleschen Variablen bilden die Menge \bar{X} :

$$g : X \rightarrow \bar{X}, x_i \mapsto \bar{x}_i \quad (5.3)$$

Die Werte von \bar{x}_i beschreiben den Zustand der entsprechenden Variablen x_i mit:

$$\bar{x}_i = \begin{cases} 0, & \text{Variable } x_i \text{ unbekannt,} \\ 1, & \text{Variable } x_i \text{ bekannt.} \end{cases} \quad (5.4)$$

$X_i \in X$ sei die Untermenge der Variablen X , welche in der Gleichung $f_i = f_i(\dots, x_j, \dots)$ vorkommen beziehungsweise gekoppelt sind. Die Kardinalität von $|X_i|$ gibt somit die Anzahl der Variablen in der Gleichung f_i an. Diese entspricht gerade dem Grad des Gleichungs-Knoten f_i im bipartiten Graph $G(X \cup F, E)$. $\bar{X}_i \in \bar{X}$ ist die entsprechende Untermenge der booleschen Variablen \bar{X} welche über $g : X_i \rightarrow \bar{X}_i$ jeweils X_i zugeordnet ist.

$F_i \in F$ sei die Untermenge der Gleichungen F , in denen die Variable x_i vorkommt. Die Kardinalität von $|F_i|$ gibt somit die Anzahl der Gleichungen an, in denen die Variable x_i erscheint. Diese entspricht gerade dem Grad des Variablen-Knoten x_i im bipartiten Graph $G(X \cup F, E)$.

Aus der Eigenschaft des bipartiten Graphens ergibt sich für $G(X \cup F, E)$:

$$\sum_i |X_i| = \sum_i |F_i|. \quad (5.5)$$

Das dynamische System soll eine diskrete, ganzzahlige Zeitentwicklung haben: $t \in \{0, 1, 2, \dots\}$ und auf dem Graph G definiert sein. Einer Gleichung f_i seien die Werte der booleschen Variablen $\bar{x}_i \in \bar{X}_i$ aus der Menge der mit ihr im Graph G verbundenen Variablen $x_i \in X_i$ bekannt. Damit lässt sich folgendes Regelwerk für die Entwicklung der Variablen \bar{X}_i der Gleichung f_i definieren:

In einem Zeitschritt $\Delta t = 1$ wird für alle Gleichungen f_i , mit $i \in \{0, 1, \dots, N\}$, die Anzahl der bekannten Variablen n_i der jeweiligen Gleichung bestimmt:

$$n_i = \sum_{\bar{X}_i} \bar{x}_i \quad (5.6)$$

Ist die Anzahl der bekannten Variablen n_i einer Gleichung f_i gerade um eins kleiner als die Zahl der in der Gleichung vorkommenden Variablen $|X_i|$, so kann die unbestimmte Variable x_j aus X_i mit $\bar{x}_j = 0$ aus f_i bestimmt werden:

$$\bar{x}_j(t+1) = 1 \text{ wenn } [x_j \in X_i \wedge \bar{x}_j(t) = 0 \wedge n_i = |X_i| - 1]. \quad (5.7)$$

Dieses Ereignis entspricht der Zuordnung der Variablen x_j zur Gleichung f_i . Dies stellt damit einen Schritt der gesuchten Lösungssequenz,

$$x_j \text{ bestimmt durch } f_i,$$

dar. Wie oben beschrieben, wird in jedem Zeitschritt das Regelwerk für jede Gleichung parallel angewandt. Die Lösungssequenz entspricht direkt der Abfolge der Zeitschritte. Somit können pro Zeitschritt der Lösungssequenz auch mehrere Lösungsschritte gleichzeitig gefunden werden. Diese Lösungsschritte können dann in der Lösungssequenz in einem Zeitschritt parallel ausgeführt werden. In der Implementierung des selbstassemblierenden Lösungspfadgenerators wird bei jedem erfolgreichen Aufruf der Regel (5.7), was einer Zuordnung von Gleichung f_i zur Bestimmung der Variablen x_j entspricht, die erfolgte Zuordnung als Lösungsschritt des jeweiligen Zeitschrittes ausgegeben und in der Liste der Lösungssequenz hinzugefügt.

5.2. Umsetzung und Implementierung

In diesem Abschnitt soll der Einsatz des zuvor abgeleiteten Regelwerkes zunächst an zwei Beispielsystemen demonstriert werden. Anschließend wird die Äquivalenz des selbstassemblierenden Lösungspfadgenerators zum konventionellen, graphenbasierten Lösungspfadgenerator demonstriert und die Laufzeiten beider Ansätze verglichen.

5.2.1. Beispiele

Einfaches Beispielsystem Als erstes Beispiel für das Auffinden einer Lösungssequenz mit dem selbstassemblierenden Lösungspfadgenerator wird das, im Sinne der Lösungspfadfindung bestimmte, Gleichungssystem (5.1) wiederverwendet. In Abbildung 5.5 ist jeder Zeitschritt des dynamischen Systems dargestellt. Die blauen Kanten stellen die Zuordnung der Gleichung f_i und der Variablen x_j eines Lösungsschrittes dar. Die Zeitvariable t in den Knoten gibt den jeweiligen Zeitschritt der Zuordnung einer Gleichung zur Variablen an. Die eingetragenen Zeitvariablen geben dann direkt die Reihenfolge der Lösungssequenz an.

In dem betrachteten Beispiel treten keine parallelen Lösungsschritte in einem Zeitschritt auf. Die Lösungssequenz in Abbildung 5.5 stimmt mit der Lösungssequenz (5.2) des graphenalgorithmischen Lösungspfadgenerators überein.

Gasturbine Als zweites Beispiel soll das Gleichungssystem der Gasturbine (2.14) aus Kapitel 2.3.6 mit dem selbstassemblierenden Lösungspfadgenerator verarbeitet werden. Dabei ergeben sich in Tabelle 9 mehrere, parallel ausführbare Lösungsschritte (vergleiche mit Ergebnis des graphenalgorithmischen Lösungspfadgenerators in Abbildung 2.14). Abbildung 5.6 zeigt den entsprechenden bipartiten Graph des selbstassemblierenden Lösungspfadgenerators mit der berechneten Lösungssequenz und der Zuordnung der Gleichungen zu den unbekannt Variablen (blaue Kanten). Die Lösungssequenzen der beiden verschiedenen Lösungspfadgeneratoren stimmen wiederum überein.

5.2.2. Äquivalenz zum graphenalgorithmischen Lösungspfadgenerator

In diesem Abschnitt soll der vorgeschlagene selbstassemblierende Lösungspfadgenerator näher analysiert und die Äquivalenz des selbstassemblierenden Lösungspfadgenerators zum graphenal-

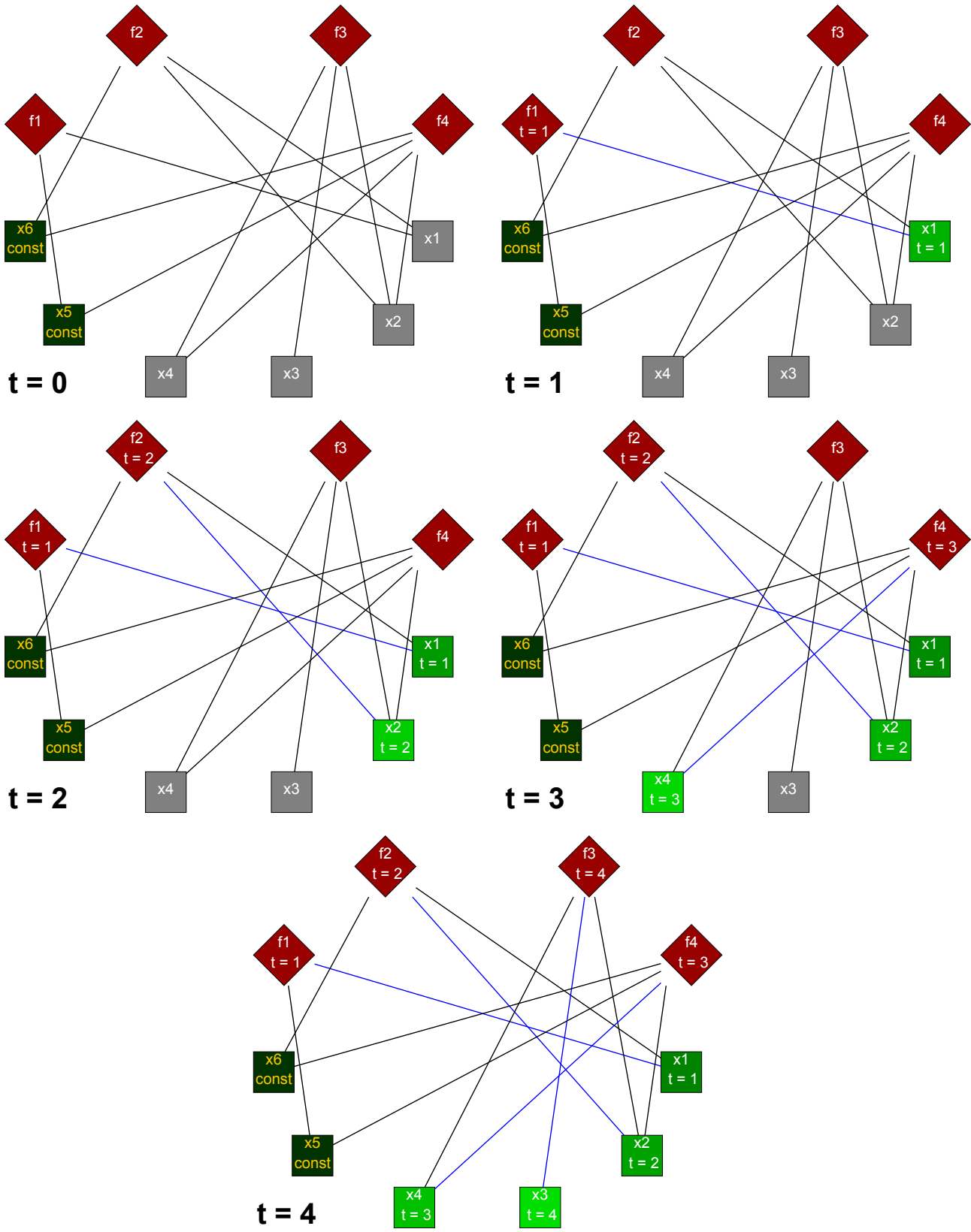


Abbildung 5.5: Zeitliche Entwicklung des selbstassemblierenden Lösungspfadgenerators dargestellt, am Beispielsystem (5.1).

Zeitschritt t	Lösungsschritte
1	$f_6 \leftrightarrow x_{12}$
2	$f_9 \leftrightarrow x_{15}$; $f_{11} \leftrightarrow x_{17}$
3	$f_7 \leftrightarrow x_6$; $f_{10} \leftrightarrow x_{18}$
4	$f_2 \leftrightarrow x_3$; $f_3 \leftrightarrow x_8$; $f_8 \leftrightarrow x_{10}$
5	$f_4 \leftrightarrow x_2$
6	$f_1 \leftrightarrow x_1$
7	$f_5 \leftrightarrow x_{11}$

Tabelle 9: Lösungssequenz des selbstassemblierenden Lösungspfadgenerators des Beispielsystems Gasturbine, bestehend aus den Gleichungen (2.14).

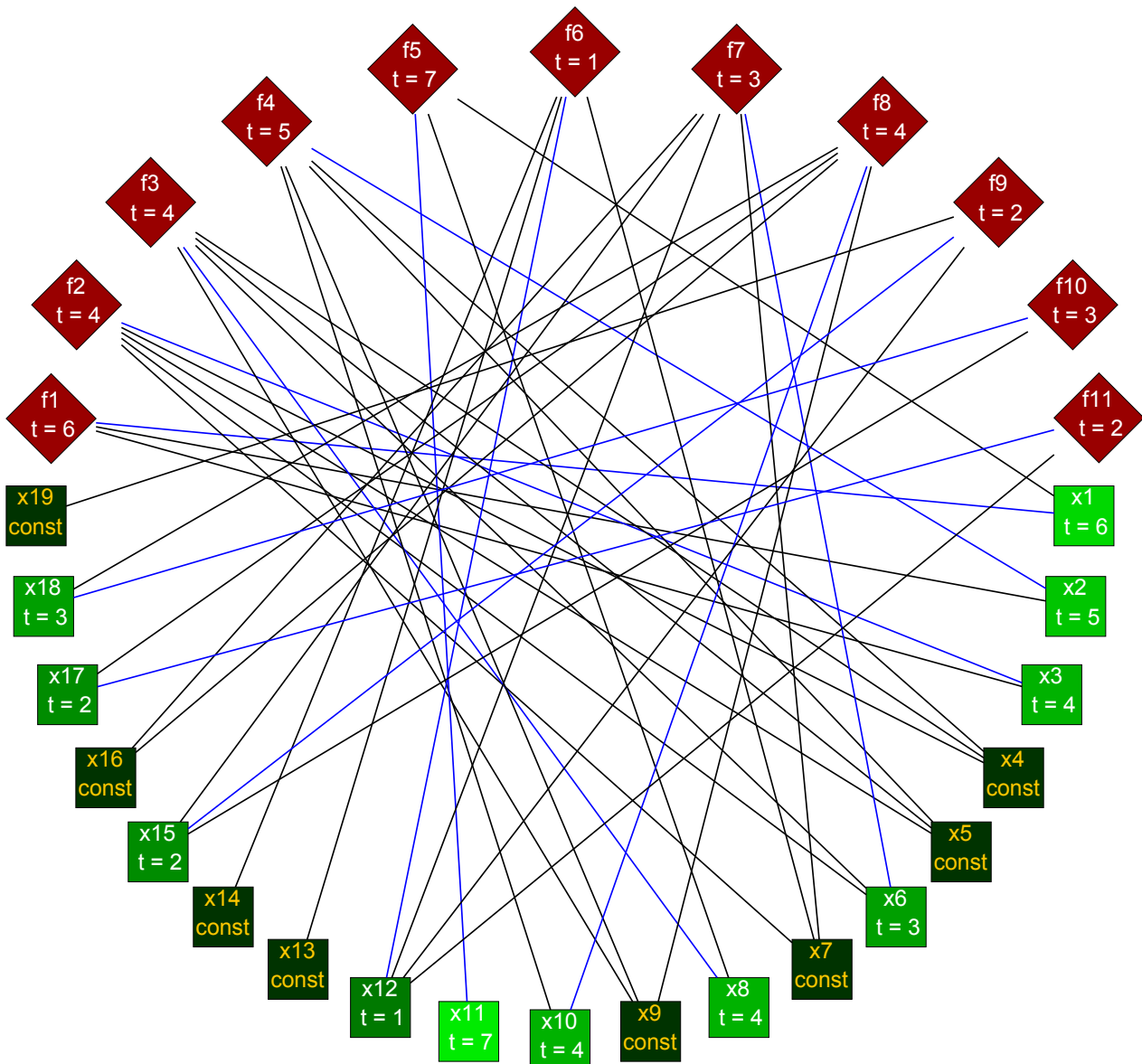


Abbildung 5.6: Finaler, bipartiter Graph ($t = 7$) des selbstassemblierenden Lösungspfadgenerators mit der berechneten Lösungssequenz und der Zuordnung der Gleichungen zu den unbekannt Variablen (blau) des Beispielsystems Gasturbine (2.14).

gorithmischen Lösungspfadgenerator aus Kapitel 2.3.6 gezeigt werden. Ein rigoroser mathematischer Beweis der Äquivalenz wird nicht durchgeführt. Eine logische Plausibilisierung soll im Rahmen dieser ingenieurwissenschaftlichen Dissertation genügen. Der Gedankengang soll am Beispielsystem der Gasturbine (2.14) illustriert werden:

Es soll der Standardanwendungsfall eines wohldefinierten Gleichungssystems betrachtet werden. Dieses besitze folgende Eigenschaften:

- Das Gleichungssystem enthält die gleiche Anzahl Gleichungen und zu bestimmender, unbekannter Variablen ($M - L = N$).
- Das Gleichungssystem zerfällt nicht in unter- oder überstimmte Teilsysteme [35, 53].
- Der Heiratssatz von Hall [77] sei für F und X erfüllt²⁷.
- Damit existiert ein perfektes Matching zwischen den Gleichungen und den unbekannt Variablen, da $|X| = |F|$.
- Das Gleichungssystem besitzt keine Strongly-Connected-Components.
- Damit existiert formal ein Lösungspfad, der alle unbekannt Gleichungen enthält.

Zunächst wird vom topologisch sortierten Graph des graphenalgorithmischen Lösungspfadgenerators ausgegangen, welcher den Lösungspfad darstellt (Abbildung 2.14). Es soll die Darstellung des Lösungspfad auf Basis der unbekannt Variablen gewählt werden (Abbildung 5.7 links).

Im ersten Schritt sollen die mit der jeweiligen unbekannt Variablen gematchten Gleichungen in den topologisch sortierten Graph wieder eingefügt werden (siehe Abbildung 5.7). Die jeweils zu einer unbekannt Variablen zugeordnete Gleichung dient dann zur Bestimmung dieser Variablen. Die gerichteten Kanten sind weiterhin so zu interpretieren, dass sie in Richtung einer Abhängigkeit zeigen: Der Knoten von dem die Kante ausgeht, soll von dem Knoten auf den die Kante zeigt abhängig sein.

Im nächsten Schritt werden die schon von Anfang an bekannten Variablen und die Abhängigkeiten der Gleichungen von diesen bekannten Variablen im rechten Graph aus Abbildung 5.7 hinzugefügt. Gleichzeitig werden die Abhängigkeiten der Gleichungen von weiteren Variablen, entsprechend dem ursprünglichen Gleichungssystem (2.14), wiederhergestellt (Abbildung 5.8).

Die Gleichungen und Variablen seien entlang der Lösungssequenz alternierend sortiert, womit die Struktur des ursprünglichen Gleichungssystems wiederhergestellt ist (vergleiche mit Abbildung 2.9). Der so wiedergewonnene Graph stellt damit eine topologisch sortierte Darstellung des ursprünglichen bipartiten Graphen, entlang der Lösungssequenz, dar.

Ausgehend von der Annahme, dass ein gültiger Lösungspfad existiert, lässt die oben dargestellte Wiedereinführung der Abhängigkeiten der Gleichungen von den bekannten Variablen die topologische Sortierung unberührt. Der gültige Lösungspfad stellt sich ja gerade als topologische Sortierung der Abhängigkeiten der unbekannt Variablen dar [171] (Abbildung 2.14).

- Bekannte Variablen werden auf das äußerste (in Abbildung 2.14 links) Ende der topologischen Sortierung platziert. Auf Grund der Tatsache, dass die Variablen bekannt sind, hängen diese nicht von anderen Gleichungen oder Variablen ab, so dass es nur Abhän-

²⁷In der graphentheoretischen Interpretation hat der Heiratssatz von Hall folgenden Inhalt: In einem Bipartiten Graph $G(X \cup F, E)$ mit Bipartition $\{X, F\}$ sind folgende Aussagen äquivalent: 1. Für alle möglichen Teilmengen der Knoten einer der Partitionen (X oder F) gilt, dass die Menge der Knoten der Teilmenge kleiner ist als die Gesamtzahl der von den Knoten der Teilmenge ausgehenden Kanten. 2. Es existiert ein (maximales) Matching das alle Knoten der betrachteten Partition (X oder F) enthält.

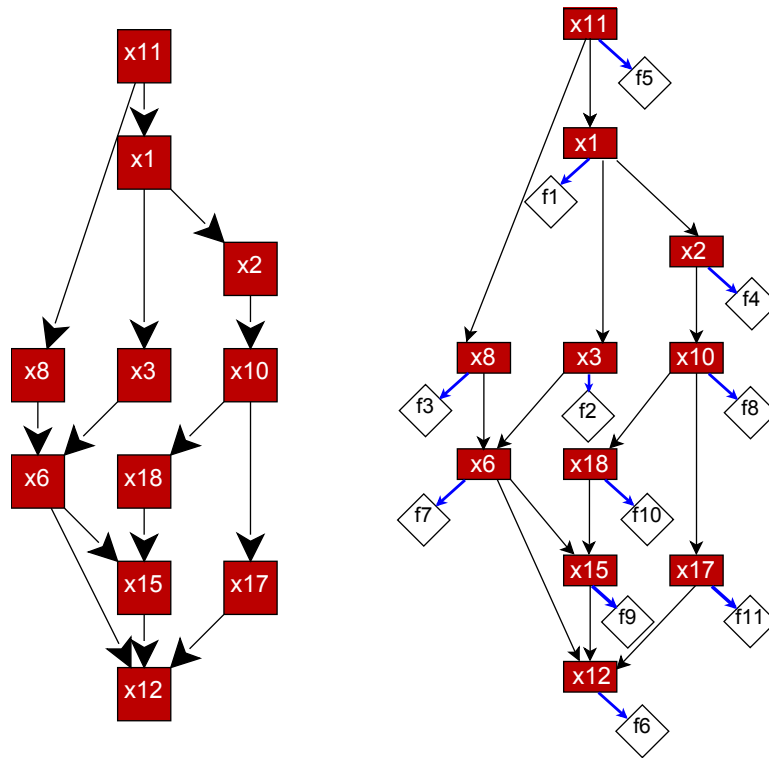


Abbildung 5.7: Linke Seite: Topologisch sortierte, unbekannte Variablen (entspricht Abbildung 2.14). Rechte Seite: Topologisch sortierte, unbekannte Variablen mit zu geordneten Gleichungen aus bipartitem Matching.

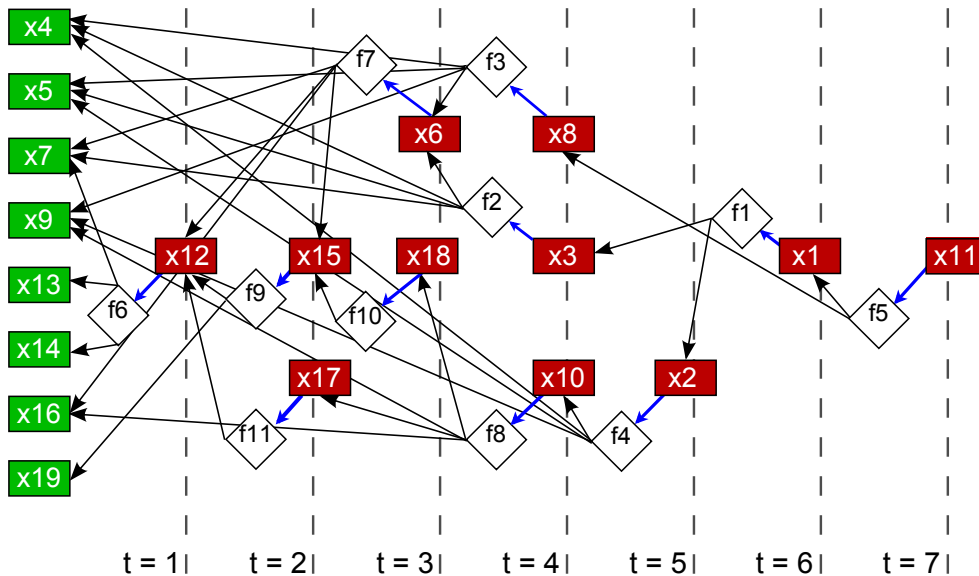


Abbildung 5.8: Topologisch sortierter Graph mit allen Abhängigkeiten des Gleichungssystems (2.14), inklusive der Zeitschritte der Lösungssequenz. Abfolge der Abarbeitung: von links nach rechts. Abhängigkeiten: von rechts nach links (=Referenzrichtung der topologischen Sortierung).

gigkeiten gibt, welche auf die bekannten Variablen zeigen und somit die topologische Sortierung intakt bleibt.

- Die im Lösungspfad (im perfekten Matching) jeder unbekanntem Variablen zugeordnete Gleichung wird direkt neben der jeweiligen Gleichung, zwischen den Sortierungsebenen der Variablenknoten, in der Referenzrichtung der topologischen Sortierung platziert. Die Abhängigkeit der unbekanntem Variablen von der gematchten Gleichung wird durch die Einführung der entsprechenden gerichteten Kante wiedergegeben, welche somit die topologische Sortierung weiterhin gewährleistet.
- Die vorher bestehenden Abhängigkeiten der jeweiligen unbekanntem Variablen von anderen Unbekanntem zeigen, auf Grund der anfangs vorliegenden topologischen Sortierung dieser Variablen, weiterhin parallel zur Referenzrichtung der Sortierung. Das Umhängen des Ausgangsendes dieser Abhängigkeitskanten, von der jeweiligen unbekanntem Variablen weg, auf den jeweils zugeordneten gematchten Gleichungsknoten der direkt benachbarten Zwischenebene der Sortierung, lässt die topologische Sortierung unberührt²⁸.

Der so gewonnene, topologisch sortierte, gerichtete, bipartite Graph des Gleichungssystems (Abbildung 5.8) soll im Folgenden für den Vergleich mit dem selbstassemblierenden Lösungspfadgenerator dienen.

Es soll noch einmal festgehalten werden, dass die vorgenommenen Umformungen, ausgehend von der Lösungssequenz des graphenalgorithmischen Lösungspfadgenerators, den anfänglichen bipartiten Graph des Gleichungssystems, bestehend aus Gleichungen F und Variablen X , wiederhergestellt haben. Der so gewonnene Graph stellt somit wieder eine umkehrbar eindeutige Abbildung des ursprünglichen Gleichungssystems dar, welcher aber nun gerichtete Abhängigkeiten enthält, entlang derer die Knoten des Graphen entsprechend der Lösungssequenz, topologisch sortiert, aufgereiht sind.

In dieser Darstellung des Gleichungssystems lässt sich die Äquivalenz zum selbstassemblierenden Lösungspfadgenerator direkt erkennen. Vergleicht man die Regeln des selbstassemblierenden Lösungspfadgenerators in Kapitel 5.1.3 (Abbildungen 5.3 und 5.4) so sind die sich ergebenden Muster identisch:

1. Gleichungs-Knoten:

Genau eine Kante ist auf eine Variable gerichtet. Die restlichen Kanten der Gleichung zeigen von der Gleichung auf andere Variablen.

2. Variablen-Knoten:

Genau eine Kante ist auf eine Gleichung gerichtet. Die restlichen Kanten der Variable zeigen von anderen Gleichungen auf die Variable selbst.

Diese Muster ergeben sich durch die oben beschriebene Wiedereinführung der Gleichungen und bekannten Variablen, welche die topologische Sortierung aber erhalten. Jeder unbekanntem Variablen ist eine Gleichung im Matching zugeordnet. Diese Zuordnung entspricht der oben erwähnten einzelnen Kante, die von der unbekanntem Variablen auf die Gleichung zeigt. Zwischen den jeweils direkt benachbarten Ebenen der unbekanntem Variablen und der Gleichungen - die Ebene der Gleichungen ist ausgehend von der Ebene der unbekanntem Variablen in Referenz-

²⁸Die neuen Ausgangsknoten der Kanten liegen weiterhin, im Vergleich zu den Endknoten, auf der entgegengesetzten Seite der Referenzrichtung der topologischen Sortierung. Der neue Ausgangsknoten der Kanten wird gerade auf der zur jeweiligen unbekanntem Variablen direkt benachbarten Zwischenebene der Sortierung gelegt, somit nur um eine "halbe" Sortierungsebene bzgl. der Variablenknoten verschoben. Dies bewirkt keine Richtungsänderung der Abhängigkeitskante.

richtung der topologischen Sortierung benachbart - existieren auf Grund der oben dargestellten Wiedereinführung der Gleichungen nur diese, dem perfekten Matching entsprechenden, Kanten.

Auf Grund der topologischen Sortierung und der bipartiten Eigenschaft des Graphens müssen nun alle anderen Kanten der unbekanntenen Variablenknoten in Referenzrichtung weg von dem Variablenknoten, auf die Gleichungsknoten gerichtet sein. Dies entspricht dem definierten Regelmuster des selbstassemblierenden Lösungspfadgenerators.

Die topologische Sortierung des Graphen führt dann direkt auf die schrittweise Dynamik des selbstassemblierenden Lösungspfadgenerators (Abbildung 5.8). Die Entwicklung des dynamischen Systems findet entsprechend entgegengesetzt der Referenzrichtung der topologischen Sortierung statt, beginnend auf der Seite der bekannten Variablen (in Abbildung 5.8 von links nach rechts). Entlang der topologischen Sortierung und der vorhandenen Muster ergibt sich, bei iterativer Regelausführung, schrittweise die gesuchte Lösungssequenz.

5.2.3. Komplexitätsbetrachtung

In einer naiven Implementierung des selbstassemblierenden Lösungspfadgenerators würde zu jedem Zeitschritt jede der Funktionen auf die Erfüllung der Regel 5.7 untersucht. Dies soll im folgendem als Regelaufruf bezeichnet werden. Im Falle des bipartiten Graphen eines Gleichungssystems $G(F \cup X, E)$ mit $|F| = |X| = N$ ergeben sich im ungünstigsten Falle N Lösungsschritte (pro Zeitschritt kann nur eine unbekannte Variable x_i bestimmt werden). Damit skaliert der Aufwand A der Regelaufufe im ungünstigsten Fall mit (siehe z.B. [112] bezüglich der Analyse und Bewertung des Aufwandes von Algorithmen):

$$\bar{A}_1 \propto O(N^2) \quad (5.8)$$

Durch die Markierung der schon zur Bestimmung einer unbekanntenen Variablen benutzten Gleichungen, kann die Zahl der Regelaufufe verringert werden. Geht man wiederum von einem Gleichungssystem mit N Lösungsschritten aus, so verringert sich die Zahl der Regelaufufe pro Zeitschritt um die im jeweils vorangegangenen Zeitschritt bestimmte Gleichung:

$$\sum_i X_i = N + N - 1 + N - 2 + \dots + 1 = \frac{(N + 1)N}{2} \quad (5.9)$$

Dies entspricht aber weiterhin einem quadratischen Aufwand in N :

$$A_1 \propto \bar{A}_1 \propto O(N^2). \quad (5.10)$$

Mit der Größe des Gleichungssystems nimmt aber auch die mögliche Zahl der Variablen $|X_i|$, die in einer Gleichung f_i vorkommen, zu. Zunächst ist eine ungünstigste Abschätzung, dass in jeder Gleichung f_i jede unbekannte Variable x_i auftritt und die Gesamtzahl der Variablen in den Funktionen damit zu:

$$\sum_i X_i = N^2 \quad (5.11)$$

wird. Dies ergibt pro Regelaufruf einen Aufwand von:

$$\bar{A}_2 \propto O(N). \quad (5.12)$$

Dieser Fall würde aber einem Gleichungssystem mit zyklischen Abhängigkeiten entsprechen, da alle Gleichungen gleichzeitig gelöst werden müssten. Ein Gleichungssystem ohne zyklische Abhängigkeiten und N Lösungsschritten kann höchstens:

$$\sum_i X_i = N + N - 1 + N - 2 + \dots + 1 = \frac{(N+1)N}{2} \quad (5.13)$$

unbekannte Variablen in den Gleichungen enthalten, da in jedem Lösungsschritt eine unbekannt Variable bestimmt wird und damit die Gleichungen für den jeweils nächsten Lösungsschritt nur von genau einer weiteren Variablen abhängen können, womit sich die arithmetische Reihe ergibt. Pro Gleichung ergibt dies im Mittel $\frac{N+1}{2}$ Variablen.

An der Aufwandsabschätzung ändert dies aber nichts:

$$A_2 \propto \bar{A}_2 \propto O(N). \quad (5.14)$$

Damit ergibt sich folgender Gesamtaufwand für die oben dargestellte Implementierung des selbstassemblierenden Lösungspfadgenerators:

$$A = A_1 \cdot A_2 \propto O(N^3). \quad (5.15)$$

Diese Aufwandsabschätzung gilt für einen sequenziellen Programmablauf. Die Regelaufrufe der einzelnen Gleichungen können aber in jedem Zeitschritt parallel ausgeführt werden. Damit reduziert sich bei der verteilten Ausführung des selbstassemblierenden Lösungspfadgenerators auf N parallelen Rechenknoten der Aufwand A_1 um den Faktor $1/N$, bei serieller Abarbeitung der Variablen X_i einer Funktion f_i innerhalb des Regelaufrufes (Gleichung (5.6)). Dies führt auf den in Gleichung (5.16) gegebenen Gesamtaufwand.

Die Verwendung eines parallelen Summationsalgorithmus (zum Beispiel der Präfix-Summen-Algorithmus [24]) für die Abarbeitung der Gleichung (5.6) ermöglicht eine weitere Reduktion des Summationsaufwandes pro Regelaufruf von $A_{2,seqSum} = A_2 = O(N)$ auf $A_{2,parSum} = O(\log(N))$ und ergibt einen noch niedrigeren Gesamtaufwand (Gleichung (5.17)).

$$A_{parSeqSum} \propto O(N^2) \quad (5.16)$$

$$A_{parParSum} \propto O(N \cdot \log(N)) \quad (5.17)$$

Im folgenden Laufzeitenvergleich (Kapitel 5.2.4) soll aber auf Grund der einfacheren Implementierung eine sequenzielle Summation mit dem in Gleichung (5.16) abgeschätzten Aufwand verwendet werden²⁹.

Der zuvor betrachtete Graph mit N Lösungsschritten hat wie oben dargestellt $E = \frac{(N+1)N}{2}$ unbekannt Variablen in den Funktionen. Dies entspricht der Zahl der Kanten E in dem entsprechenden bipartiten Graph des Gleichungssystems. Damit lässt sich die Performance direkt mit den in Tabelle 1 aufgetragenen Aufwänden der Graphenalgorithmus vergleichen. Der Aufwand für das maximale bipartite Matching bestimmt den Aufwand des graphenalgorithmischen Lösungspfadgenerators, da dieser polynomial skaliert und die anderen Graphenalgo-

²⁹Durch den Übergang auf eine problemangepasste Hardwarestruktur, z.B. einen Analogrechner, wird eine gleichzeitige Aufsummierung von mehreren Eingangssignalen möglich. Dies würde erlauben, die bisher angenommene Abarbeitung der Variablenliste X_i im Regelaufruf einer Funktion f_i (Gleichung (5.6)), in einem Schritt zu bewerkstelligen und führt so auf einen Gesamtaufwand von $O(N)$.

rithmen des Lösungspfadgenerators linear im Aufwand skalieren. Setzt man $E = \frac{(N+1)N}{2}$ und $V = 2 \cdot N = |X| + |F|$ so ergibt sich je nach Matching-Algorithmus folgender Aufwand:

$$\begin{aligned} A_{Hopcraft-Carp} &\propto O(N^{5/2}) \\ &\text{beziehungsweise} \\ A_{AltBlMePa} &\propto O\left(\frac{N^{5/2}}{\log(2N)}\right). \end{aligned} \quad (5.18)$$

Damit weist die parallelisierte Implementierung des selbstassemblierenden Lösungspfadgenerators den besten Aufwandswert auf. Die sequenzielle Berechnung des selbstassemblierenden Lösungspfadgenerators weist in der Komplexitätsanalyse dagegen höhere Aufwände auf als der aufwandsbestimmende Schritt des graphenalgorithmischen Lösungspfadgenerators.

5.2.4. Laufzeitenvergleich

Vergleicht man beide Algorithmen in der Praxis, so zeigen sich im Folgenden erhebliche Laufzeitenvorteile für den selbstassemblierenden Lösungspfadgenerator. Für den Laufzeitenvergleich wurde eine parallelisierte Version des selbstassemblierenden Lösungspfadgenerators implementiert. Wie bereits erwähnt, sollen dafür die Gleichungen auf die zur Verfügung stehenden Rechenkerne verteilt werden. Die Abarbeitung der Variablenliste X_j im Regelaufruf einer Funktion f_j nach Gleichung (5.6) findet sequenziell statt³⁰.

Auf Grund der sequenziellen Abarbeitung der Variablenlisten, der einem Kern i zugeordneten Funktionen f_j , wird eine, bezüglich der Gesamtzahl der Variablen des Kernes i

$$|X_{CPU_i}| = \sum_{\{j|f_j \rightarrow CPU_i\}} |X_j|, \quad (5.19)$$

gleichmäßige Verteilung der Funktionen über die zur Verfügung stehenden Kerne angestrebt:

$$|X_{CPU_1}| \approx \dots \approx |X_{CPU_i}| \approx \dots \approx |X_{CPU_L}|. \quad (5.20)$$

Diese Verteilung verspricht, unter der Annahme, dass die Abarbeitung der Variablenliste der Funktionen innerhalb eines Regelaufrufes den zeitbestimmenden Faktor darstellt, eine synchrone Ausführung zwischen den Kernen ohne Laufzeitdifferenzen. Eine automatisierte Verteilung der Funktionen auf die verfügbaren Rechenkerne kann durch die Implementierung entsprechender Graphenpartitionierungsalgorithmen [55, 39] oder die Verwendung von Parallelisierungsbibliotheken, zum Beispiel METIS [101] oder SCOTCH [40], dargestellt werden.

Zwei Gleichungssysteme sollen mit dem selbstassemblierenden Lösungspfadgenerator sowie mit einer auf Graphenalgorithm basierenden Vergleichsimplementierung des Design Compiler 43V2 [90] für verschiedene Gleichungssystemgrößen M verarbeitet werden. Das erste Gleichungssystem (5.21) sei ein dünn besetztes Gleichungssystem:

$$\begin{aligned} &f_1(x_1) \\ &f_2(x_1, x_2) \\ &f_3(x_2, x_3) \\ &\quad \vdots \\ &f_M(x_{M-1}, x_M) \end{aligned} \quad (5.21)$$

³⁰Eine Parallelisierung dieser Summation würde die Implementierung wesentlich verkomplizieren und zusätzliche, auf der verwendeten Hardware nicht verfügbare, Rechenkerne benötigen.

Für die parallelisierte Berechnung werden die Funktionen in gleich großen, aneinanderhängenden Partitionen nach aufsteigendem Index auf die L CPUs verteilt (f_0 bis $f_{\frac{M}{L}-1}$ auf $CPU = 1$, $f_{\frac{M}{L}}$ bis $f_{\frac{2M}{L}-1}$ auf $CPU = 2$, usw.). Dies gewährleistet die beschriebene Gleichverteilung der Variablen auf die CPUs nach Gleichung (5.20).

Das zweite Gleichungssystem (5.22) entspricht dem oben erwähnten, am dichtesten besetzten Gleichungssystem mit M Lösungsschritten, welches keine zyklischen Abhängigkeiten aufweist und eine Dreiecksform besitzt:

$$\begin{aligned}
 & f_1(x_1) \\
 & f_2(x_1, x_2) \\
 & f_3(x_1, x_2, x_3) \\
 & \vdots \\
 & f_M(x_1, x_2, \dots, x_{M-1}, x_M)
 \end{aligned} \tag{5.22}$$

Das dicht besetzte System wird für die parallelisierte Berechnung alternierend, nach aufsteigendem Index der Gleichung f_i auf die L CPUs verteilt (f_i auf $CPU = i \bmod L$), um die gleichmäßige Verteilung über die Kerne nach Gleichung (5.20) zu erreichen.

Die Tabellen 10 und 11 enthalten die Laufzeiten der jeweiligen Gleichungssysteme, für verschiedene Gleichungssystemgrößen M und Prozessorkernzahlen, der beiden Lösungspfadgeneratoren im direkten Vergleich. Die Algorithmen sind jeweils in Java programmiert und wurden auf einem handelsüblichem PC durchgeführt und gemessen.

Gleichungssys. Größe $M = F $	Graphenalgorithm. Lösungspfadgenerator CPUs=1	Selbstassemblierender Lösungspfadgenerator		
		CPUs=1	CPUs=2	CPUs=4
10	1 ms	1 ms	2 ms	3 ms
100	5 ms	5 ms	9 ms	14 ms
1000	281 ms	142 ms	100 ms	100 ms
10000	32844 ms	13753 ms	6279 ms	4300 ms
25000	302469 ms	88707 ms	43055 ms	27274 ms
50000	1452437 ms	345800 ms	181025 ms	108451 ms

Tabelle 10: Laufzeiten der Lösungspfadgeneratoren für das dünn besetzte System (5.21).

Gleichungssys. Größe $M = F $	Graphenalgorithm. Lösungspfadgenerator CPUs=1	Selbstassemblierender Lösungspfadgenerator		
		CPUs=1	CPUs=2	CPUs=4
10	1 ms	1 ms	1 ms	3 ms
100	47 ms	16 ms	11 ms	14 ms
1000	30562 ms	11312 ms	5949 ms	2916 ms
2500	-*	175986 ms	86334 ms	44550 ms
5000	-*	1401067 ms	689939 ms	365305 ms

Tabelle 11: Laufzeiten der Lösungspfadgeneratoren für das dicht besetzte Gleichungssystem (5.22). Die mit * gekennzeichneten Konfigurationen konnten auf Grund der Arbeitsspeicheranforderungen nicht berechnet werden.

Der Vergleich der Laufzeiten zwischen dem graphenalgorithmischen und dem selbstassemblierenden Lösungspfadgenerator zeigt, dass bereits die serielle Implementierung und Ausführung des selbstassemblierenden Lösungspfadgenerators (CPUs=1) erhebliche Laufzeitenvorteile aufweist. Dies könnte an den einfacheren Datenstrukturen und Operationen des selbstassemblierenden Lösungspfadgenerators liegen, bei dem keine komplexen Operationen auf Graphen durchgeführt werden müssen.

Ein Vorteil der parallelisierten Version des selbstassemblierenden Lösungspfadgenerators ist erst für Gleichungssysteme mit mehr als einigen hundert Gleichungen zu erkennen, wenn die Beschleunigung durch Parallelisierung von Operationen den Zusatzaufwand der Parallelisierung überwiegt. Der implementierte selbstassemblierende Lösungspfadgenerator übertrifft die graphenalgorithmische Vergleichsimplementierung des Design Compiler 43V2 [90] in der berechenbaren Gleichungssystemgröße.

In der Anwendung des selbstassemblierenden Lösungspfadgenerators darf aber nicht außer Acht gelassen werden, dass wohldefinierte Gleichungssysteme ohne zyklische Abhängigkeiten vorausgesetzt wurden. Wird ein Gleichungssystem mit zyklischen Abhängigkeiten im selbstassemblierenden Lösungspfadgenerator eingesetzt, so kann dieser nicht alle unbekanntenen Variablen bestimmen. In diesem Falle kann auf die Hilfskonstruktion des graphenalgorithmischen Lösungspfadgenerators zurückgegriffen werden, welcher zyklische Abhängigkeiten detektieren kann und in einen Superknoten überführt. Wie bereits erwähnt, stellt dieser Fall in der Entwurfssprachenpraxis aber eine Ausnahme dar.

6. Dimensionsabhängige Entwurfsabfolge

In diesem Kapitel sollen die in den vorherigen Kapiteln dargestellten Ergebnisse im Licht einer allgemeingültigen Entwurfsmethodik betrachtet werden. Dazu wird zunächst das Zusammenspiel der Entwurfssprache in Kapitel 3.1 mit der Verrohrungssprache von Kapitel 4.2 betrachtet. Hierbei wird zunächst das Entwurfsproblem, in Anlehnung an das Constraint-Satisfaction-Problem, als das Auffinden der Schnittmenge von gültigen Teilentwürfen aufgefasst, welche selbst wiederum Teilmengen eines einbettenden mathematischen Entwurfsraumes sind.

Die mathematischen Dimensionen dieser, gültige (Teil-)Entwürfe repräsentierenden, Teilmengen werden analysiert. Aus den mathematischen Dimensionen dieser Teilentwürfe lässt sich anschließend auf die Abfolge einzelner Entwurfsschritte schließen. Daraus können Regeln für die systematische Bestimmung der Entwurfssequenz abgeleitet werden.

6.1. Lösungsweg und Methodik

In diesem Abschnitt werden zunächst formale Aspekte der Entwurfsbeschreibung dargestellt. Am Beispiel des Zusammenspiels von Packaging und Routing, der in dieser Arbeit erstellten Gesamtentwurfssprache, wird die Bedeutung der mathematischen Dimensionen der Teilentwürfe für den Entwurfsablauf abgeleitet.

6.1.1. Entwurfsprozess als Nichtlineares System

Das den Entwurfsprozess abbildende Produktionssystem, welches beispielsweise über Java-Regeln oder Entscheidungsknoten³¹ vom Zustand des Entwurfsgraphen abhängige Aktivitäten- und Regelabfolgen enthält, stellt ein nichtlineares System dar.

Die Abbildung einer Entwurfsentscheidung im Entwurfsprozess (if-then-Schema) in einem Entscheidungsknoten oder einer Java-Regel stellt prinzipiell nichts anderes als die in Kapitel 2.5.1, als Beispiel eines nichtlinearen Systems, dargestellte Sprungfunktion dar: In Abhängigkeit der Eingangs- oder Bewertungsgrößen x wird eine von zwei möglichen Regeln $y = 0$ oder $y = 1$ im Entwurfsprozess aufgerufen. Damit einher geht eine empfindliche Abhängigkeit der Regelabfolge (Ausgangssignal) von den Eingangsgrößen x (Anfangswerte) und damit eine empfindliche Abhängigkeit des finalen Entwurfes von den Anforderungen und Randbedingungen.

Ein weiteres Beispiel nichtlinearen Verhaltens stellt die Geometrie der Verrohrungssprache dar. Die Synthese der Verrohrung soll als (Sub-)System des Entwurfsprozesses betrachtet werden. Das Subsystem soll eine vorgegebene Rohrbogenkonfiguration I , die Startposition \vec{P}_S und Startrichtung \vec{D}_S , die Endposition \vec{P}_E und Endrichtung \vec{D}_E sowie den Bauraum als Eingangssignale aufnehmen. Das Ausgangssignal y soll wiederum zwei Werte annehmen können: Ein Wert für die Existenz einer mit den Randbedingungen kompatiblen Verrohrung. Den anderen Wert, wenn keine Verrohrung unter den gegebenen Randbedingungen existiert:

$$y = f(\vec{P}_E, \vec{D}_E)_{I, \vec{P}_S, \vec{D}_S} = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases} \quad (6.1)$$

Betrachtet man die Zielmenge der Verrohrungssprache in Abbildung 4.19 von Kapitel 4.2.3 so wird klar, dass sich die Übertragungsfunktion (6.1) beziehungsweise die Lösungsmenge der Verrohrung im Allgemeinen nichtlinear verhalten.

Dieses nichtlineare Verhalten mit sensitiver Abhängigkeit der Systemantwort bzw. der Existenz eines gültigen Entwurfes von Eingangsgrößen, im Entwurf Randbedingungen und Anforderungen, impliziert, dass bei der Änderung eines Entwurfes beziehungsweise der Eingangsgrößen die abhängigen Systeme jeweils neu ausgelegt und entworfen werden müssen³².

Daraus folgt im geometrisch manifestierten Entwurf die Notwendigkeit nach einem vollständigen Durchlaufen des Entwurfsprozesses nach einer Änderung der Eingangsgrößen des Entwurfes, da die geometrisch manifestierten (Sub-)Systeme alle im Konfigurationsraum (realer Raum, in dem das Produkt zusammengebaut wird) prinzipiell miteinander verkoppelt sind. Für einen abgeänderten Entwurf muss eine gegenseitige Durchdringung der geänderten Komponenten überprüft werden, welche sich im schlimmsten Fall auf alle weiteren (Sub-)Systeme fortpflanzt.

Hierarchische Konzepte, welche den Sub-Komponenten a priori Teilvolumina des Konfigurationsraumes zuweisen, beziehungsweise von einer anfänglichen Zuweisung der Position und des Raumes der noch nicht detaillierten Einzelkomponenten [138, 60] ausgehen, könnten hier Abhilfe schaffen (Divide and Conquer [43]). Damit kann eine Entkopplung der geometrischen Abhängigkeiten erreicht werden, wenn die jeweiligen Systeme im Konfigurationsraum nicht direkt miteinander geometrisch verbunden sind oder beliebige Verbindungsteile erst nachträglich

³¹Dies kann unmittelbar auf Entwurfsentscheidungen im Entwurfsprozess, unabhängig von der verwendeten Modellierungssprache, verallgemeinert werden: Der in [138] dargestellte Entwurfsprozess (Abb. 4.3. in [138]: "Hauptarbeitsschritte beim Planen und Konzipieren") besteht, vor allem während der konzeptionellen Phase, jeweils aus einer Suche möglicher Lösungsprinzipien oder Konzepte, mit einer anschließenden Auswahl (*Entscheidung*) für eine oder mehrere der Möglichkeiten.

³²Es wird davon ausgegangen, dass keine a priori Information über den jeweiligen Entwurf vorliegt.

erzeugt werden können [54]. Letzteres ist aber im Allgemeinen nicht möglich, da nicht für beliebige gegenseitige Positionierungen der Einzelsysteme geeignete, (physikalisch) ausreichend dimensionierte Verbindungsteile garantiert werden können.

6.1.2. Entwurfsraum und Entwurf als Constraint-Satisfaction-Problem

Der im letzten Abschnitt dargestellten Betrachtung der Zielmenge der Verrohrung aus Kapitel 4.2.3 folgend, soll der Prozess des Entwerfens, als Zusammensetzen von Subsystemen, noch einmal näher betrachtet werden. Dazu sind in Abbildung 6.1 die, in [113] auf Entwurfssprachen bezogenen, verschiedenen Ebenen der Sprachbeschreibung dargestellt (Syntax, Semantik, Pragmatik). Die Syntax der Entwurfssprache wird durch die Menge der aus den erlaubten Regelkombinationen generierbaren Entwürfe gebildet [113] (kombinatorisch mögliche Entwürfe). Die semantisch korrekten Sätze bzw. Entwürfe, welche eine Untermenge der syntaktisch korrekten Entwürfe bilden, stehen für Entwürfe, die eine "inhaltliche Bedeutung" besitzen [113] und in diesem Sinne technisch sinnvoll sind, beispielsweise keine geometrischen Durchdringungen aufweisen. In die Menge der semantisch gültigen Entwürfe ist wiederum die Menge der pragmatisch korrekten Entwürfe als Untermenge eingebettet, welche in einer speziellen Situation mit gegebenen Randbedingungen und Anforderungen kompatibel ist [113].

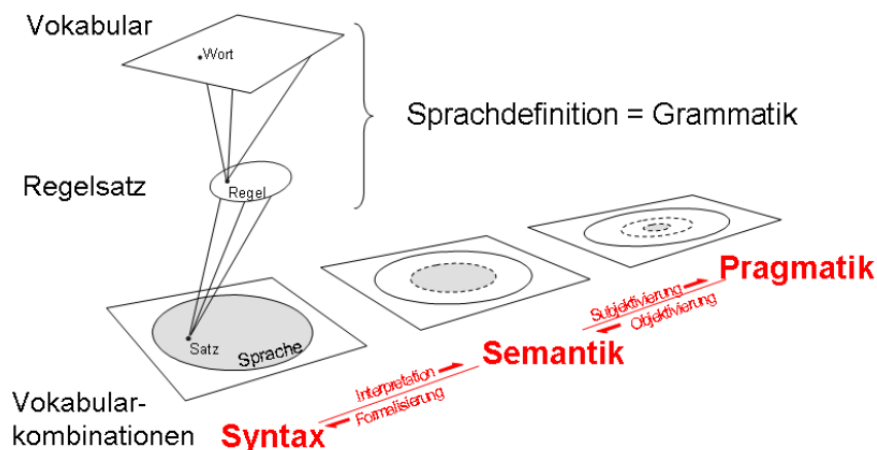


Abbildung 6.1: Ebenen der Sprachbeschreibung formaler Sprachen am Beispiel von Entwurfssprachen, nach Rudolph entnommen aus [113].

Diese Einteilung des Entwurfes, anhand formaler Sprachkonzepte in verschiedene Ebenen der Sprachbeschreibung, kann um eine weitere Ebene nach unten erweitert werden (Abbildung 6.2): Das Alphabet, welches die symbolische Sprachrepräsentation darstellt, auf der die oben genannten Beschreibungsebenen aufbauen, wird vom Autor als Ergänzung vorgeschlagen. Diese unterste Ebene stellt die elementaren Bausteine zur Generierung der Entwürfe zur Verfügung und bildet so die Basis zur Definition von Entwurfsregeln.

Betrachtet man den Entwurf, zumindest den mathematisch formulierbaren Teil, als Constraint-Satisfaction-Problem entsprechend Kapitel 2.2.6, so lassen sich den verschiedenen Beschreibungsebenen der Entwurfssprachen formal analoge Definitionen des Constraints-Problems zuordnen. Es soll wiederum die in Kapitel 2.2.6 dargestellte, weitergefasste und pragmatische Definition des Constraint-Problem in der Entwurfsanwendung gelten.

Linguistik	Entwurf(-sprachen)	Constraints	Anforderungen bzw. $ C $	DoF	Entwurfsraum
Pragmatik	situativ optimal (unter gegebenen Anforderungen & Randbedingungen)	$\langle X, D, C_S + C_P \rangle$	↑	↓	(finale) Entwürfe
Semantik	technisch sinnvoll (z.B. keine Durch- dringungen)	$\langle X, D, C_S \rangle$			(expliziter) Sub- Entwurfsraum
Syntax	kombinatorisch möglich (Vokabeln und Regeln)	$\langle X, D, \emptyset \rangle$			(impliziter) Entwurfsraum
Alphabet	Entitäts-Beschreibung (Vokabeln)	$\langle X, \emptyset, \emptyset \rangle$			mathematischer Entwurfsraum

Abbildung 6.2: Entwurfsraumdefinitionen und Analogien ausgewählter Entwurfs- und Beschreibungsparadigmen für verschiedene Sprachbeschreibungsebenen. Die ersten beiden Spalten folgen [159]. Die Spalte *Constraints* bezieht sich auf die Definition des Constraints-Problem aus Kapitel 2.2.6. Die Unterscheidung von *explizitem* und *implizitem* Entwurfsraum lehnt sich an [207] an.

Auf der untersten Stufe des Alphabets soll das zugeordnete Constraint-Problem nur aus der Menge der Variablen X bestehen. Die Variablen spannen den einbettenden Raum des Constraint-Problems auf.

Auf der Ebene der Syntax der Entwurfsgrammatik bestehe das zugeordnete Constraint-Problem aus den Variablen X und einer zugeordneten Wertemenge D . Die Wertemenge soll die Werte der Variablen formal auf die dem Regelwerk entsprechenden, kombinatorisch möglichen Wertepaare der Variablen einschränken³³.

In der nächst höheren Ebene (Semantik) wird diese kombinatorische Vielfalt auf die sinnvollen Kombinationen eingeschränkt. Dies kann durch die Definition einer ersten Menge von Zwangsbedingungen (Constraints) C_S im Constraint-Satisfaction-Problem formal erreicht werden.

Die oberste Stufe (Pragmatik), welche die mit den Anforderungen und den Randbedingungen optimale Entwürfen enthält, kann im Constraint-Problem durch das Hinzufügen dieser zusätzlichen Zwangsbedingungen C_P erreicht werden.

Die Spalten *DoF* beziehungsweise Anforderungen ($|C|$ entspricht der Zahl der Zwangsbedingungen) stellen die Zahl der Freiheitsgrade beziehungsweise die Zahl der Randbedingungen oder Anforderungen schematisch dar. Beide Größen sind gegenläufig: Eine höhere Zahl von Anforderungen, beziehungsweise die Eingrenzung des Entwurfsraumes auf die situativ optimalen Entwürfe (Pragmatik) im Entwurfsprozess, entspricht einer Reduktion der Freiheitsgrade.

Die mit den verschiedenen Sprachbeschreibungsebenen einhergehende Einschränkung der Freiheitsgrade kann auch auf eine Klassifizierung des Entwurfsraumes übertragen werden (rechte Spalte in Abbildung 6.2): Der auf einer Sprachebene zugeordnete Entwurfsraum stellt jeweils

³³Diese Zuordnung soll schematischer Natur sein. Im Allgemeinen ist nicht garantiert, dass die Menge der syntaktisch korrekten Entwürfe nur über die Wertemenge des Constraint-Satisfaction-Problems abgebildet werden kann, sondern zur Abbildung ggf. einen ersten Satz von Zwangsbedingungen C_0 erfordert.

die Menge der Entwürfe dar, welche die Definition der Sprachebene (syntaktische, semantische und pragmatische Korrektheit), beziehungsweise die dazu analogen Randbedingungen des Constraint-Problems erfüllen.

Die oberste Ebene soll die Bereiche des Entwurfsraumes enthalten, welche situativ optimale Entwürfe, unter Einhaltung der gegebenen Anforderungen und Randbedingungen, umfassen. Diese entspricht der Sprachbeschreibungsebene der Pragmatik.

Die angedeutete Einteilung der Ebenen *Semantik* und *Syntax* in impliziten und expliziten Entwurfsraum lehnt sich an Woodburry [207] an. Nach [207] würden die Regeln von Entwurfsgammatiken im idealen Fall ausschließlich und vollständig alle semantisch korrekten Entwürfe erzeugen. Das vollständige, ausschließliche Ausfüllen der semantischen Einhüllenden des Entwurfes kann aber im Allgemeinen bei der Definition eines Regelwerks nicht a priori garantiert werden. Dies erfordert im Allgemeinen eine Analyse bezüglich der semantischen Korrektheit eines synthetisierten Entwurfes.

Diese mit einem Regelwerk beziehungsweise einer Grammatik erzeugbaren Entwürfe sind somit zunächst einmal der Ebene der Syntax zuzuordnen, da die semantische Korrektheit, wie oben dargestellt, im Allgemeinen nicht a priori garantiert werden kann. Dieser Entwurfsraum, der alle generierbaren Entwürfe umfasst, soll nach Woodburry als *impliziter* Entwurfsraum bezeichnet werden [207]. Entsprechend ordnet Woodburry bereits bekannte und wiederverwendbare Entwürfe, welche technisch sinnvolle Entwürfe darstellen, dem *expliziten* Entwurfsraum zu. Diese sind damit in der Ebene der *Semantik* anzusiedeln.

Der mathematische Raum, in dem die regelbasiert erzeugten und kombinatorisch möglichen Entwürfe als Untermengen eingebettet sind, soll als mathematischer Entwurfsraum bezeichnet werden. Dieser stellt damit, als einbettender Raum, die zur Beschreibung der erzeugbaren Entwürfe notwendigen Freiheitsgrade zur Verfügung und weist eine entsprechende mathematische Dimension auf.

Die Abbildung 6.3 zeigt ein schematisches Beispiel des Entwurfsraumes für die verschiedenen Beschreibungsebenen von Abbildung 6.2. Das betrachtete Beispiel stellt den Einbau eines Bauteils (rotes Quadrat) in einem zweidimensionalen Bauraum dar, in dem sich schon ein anderes Bauteil (graues Quadrat) befindet. Die jeweiligen Einbaupositionen (rosa) des roten Quadrates beziehen sich auf den Mittelpunkt des einzubauenden Quadrates und stellen die Entwurfsräume der verschiedenen Beschreibungsebenen dar.

Der mathematische Entwurfsraum (Alphabet) wird vom entsprechenden zweidimensionalen Konfigurationsraum gebildet, der die Basis für die Abbildung der Entwurfsgrößen bildet. Die kombinatorisch möglichen Einbaupositionen (Syntax) bei einer Ausführung einer Einbauregel entsprechen der Menge aller möglichen Punkte im Entwurfsraum. Dies ist in der Spalte *b)* von Abbildung 6.3 angedeutet. Diese Positionen beinhalten auch technisch unsinnige Einbauten, wie die angedeutete Kollision des bestehenden Bauteils mit dem einzubauenden Teil.

Die technisch sinnvollen Einbaupositionen (Semantik) sind in Spalte *c)* der Abbildung 6.3 angedeutet. Eine über einen direkten Kontakt realisierte Befestigung der beiden Bauteile soll einen technisch sinnvollen Einbau darstellen. Die Menge dieser Einbaupositionen stellt den Umriss eines Quadrates dar.

Auf der Ebene der Pragmatik seien die Randbedingungen und Anforderungen an den situativ optimalen Entwurf so, dass das neue Bauteil möglichst in positiver x_2 -Richtung eingebaut werde und die Länge der gemeinsamen Kante beider Bauteile maximal sein soll. Die dann

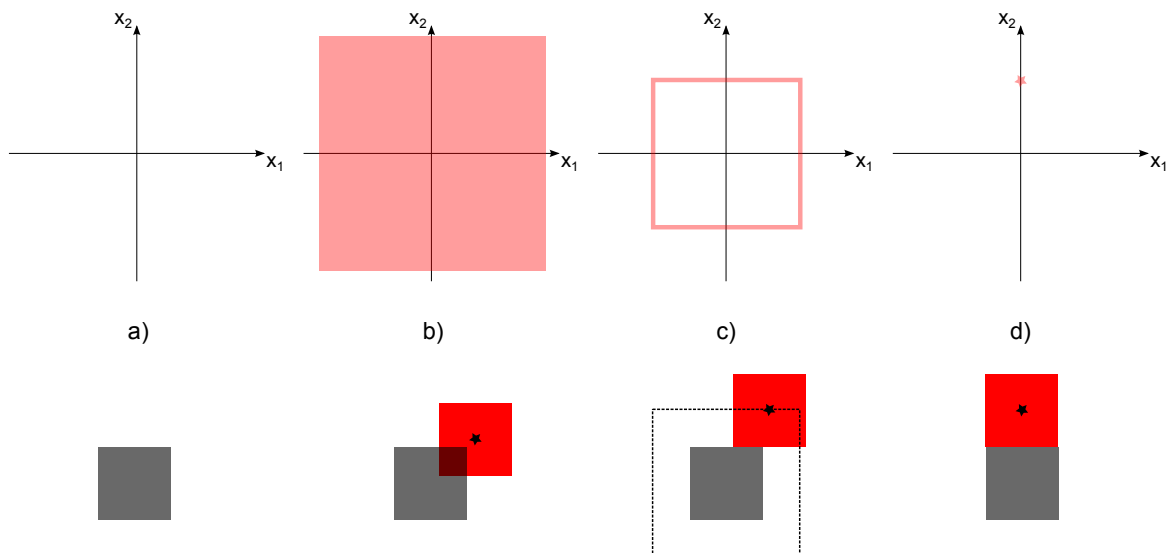


Abbildung 6.3: Untere Hälfte: Einbau eines Bauteiles (rotes Quadrat) in einen ebenen Bau- raum mit bereits eingebautem Bauteil (graues Quadrat). Der entsprechende Entwurfsraum ist in der oberen Hälfte dargestellt: Von links nach rechts: a) mathematischer Entwurfsraum umfasst die Variablen des Entwurfsproblems, b) kombinatorisch mögliche Einbaupositionen (rosa, bezogen auf Mittelpunkt des roten Quadrates), c) sinnvolle Einbaupositionen ohne Durchdringung, d) optimale Einbauposition.

übrigbleibende, situativ optimale Einbauposition ist in Spalte *d*) als Punkt im mathematischen Entwurfsraum dargestellt.

Gut zu erkennen ist, dass die Freiheitsgrade des Entwurfes, bei steigender Zahl der Anforderungen und Randbedingungen, abnehmen. Im dargestellten Beispiel kommt es zu einer sukzessiven Reduktion der Dimensionen, der, nach Abbildung 6.2 entlang der Sprachbeschreibungsebenen definierten, Entwurfsunterräume bzw. der die Entwürfe repräsentierenden Teilmengen des mathematischen Entwurfsraumes (*b*) \rightarrow Fläche; *c*) \rightarrow Linie; *d*) \rightarrow Punkt).

6.1.3. Reihenfolge von Packaging und Routing

Auf der im vorherigen Abschnitt verfolgten Darstellung des Entwurfsraumes aufbauend, soll das Zusammenspiel der Bestimmung der Einbaupositionen aus Kapitel 3.1 und 3.2 noch einmal näher betrachtet werden: In der in Kapitel 3.1 dargestellten Entwurfssprache werden in der Abfolge des Entwurfsprozesses zuerst die Einbaupositionen der Komponenten bestimmt (Packaging) und anschließend die Verrohrung, basierend auf den Aus- und Einlasspositionen der im Packaging positionierten Komponenten, synthetisiert (Routing).

Die Verrohrung des Kamines (Abbildung 3.34 bzw. Abbildung 6.4 links) erscheint schon als brauchbar im Rahmen eines Konzeptentwurfes. Trotzdem stellt diese Anordnung bei näherer Betrachtung noch nicht das Optimum bezüglich maximaler Einfachheit (KISS-Prinzip [148]) des Entwurfes dar. So würde beispielsweise die Anordnung der Katalysatorboxen und der Mischstrecken entlang einer gemeinsamen Achse einen noch einfacheren Entwurf, mit ausschließlich geraden Rohrelementen zwischen den genannten Komponenten (Abbildung 6.4 rechts), ergeben. Dies würde einer leeren Rohrbogenkonfiguration $I = \{\} = \emptyset$ entsprechen.

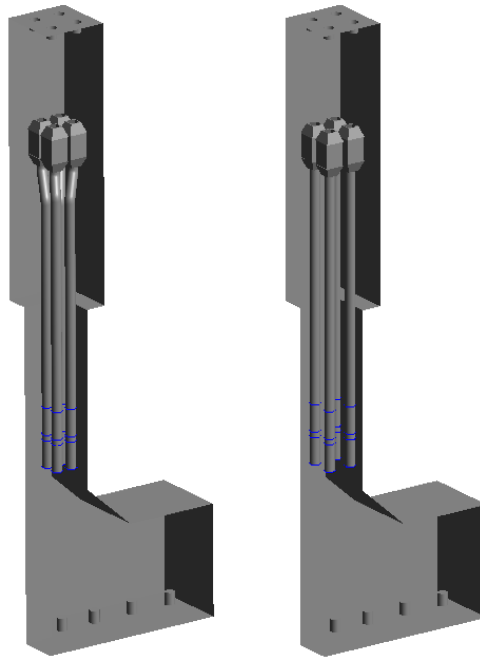


Abbildung 6.4: Linke Seite: Verrohrung des Abgaskamines mit Hilfe der Entwurfssprachen aus den Kapiteln 3.1 und 3.2. Rechte Seite: Einfachste Verrohrung des Abschnittes zwischen Mischstrecken und SCR-Katalysatorboxen (gerade Rohre) bei axialer Anordnung der Komponenten.

Diese einfachste Konfiguration wurde aber auf Grund der sequenziellen Ausführung der Einbaupositionsbestimmung von Mischstrecken, SCR-Katalysatorboxen sowie der nachfolgenden Synthese der Verrohrungen nicht gefunden. Für das Auffinden der Konfiguration mit geraden Rohrstrecken hätten schon in der vor der Verrohrung ausgeführten Bestimmung der Einbaupositionen die Zwangsbedingungen dieser einfachsten Verrohrung bekannt sein müssen. Dies könnte entweder über die *gleichzeitige* Ausführung der Einbaupositionsbestimmung (Packaging) und der Verrohrungssynthese (Routing) realisiert werden, was aber einer funktionalen Entkoppelung der einzelnen Aufgaben des Entwurfes zuwiderläuft, oder über ein Vertauschen der Ausführungsschritte von Packaging und Routing dargestellt werden.

In der Abbildung 6.5 oben sind die Teilmengen (rosa) des oben betrachteten Packaging (links) und Routing (rechts) eines Rohrabschnittes, welche bei isolierter Betrachtung der jeweiligen Entwurfsaufgabe technisch sinnvolle Entwürfe repräsentieren, schematisch im mathematischen Entwurfsraum dargestellt. Aus Gründen der Darstellung soll nur eine Rohrlinie im gegebenen Bauraum (schwarz konturierter Quader) betrachtet werden. Im Routing wird von der, in Abbildung 6.4 rechts dargestellten, geraden Rohrstrecke ausgegangen.

Um die Anzahl der darzustellenden Freiheitsgrade übersichtlich zu gestalten, wird davon ausgegangen, dass der Startpunkt \vec{P}_S und die Startrichtung \vec{D}_S der Verrohrung fest vorgegeben sind (blau dargestellt). Im Packaging soll nun die Einbauposition der Komponente SCR-Box bestimmt werden, welche an das Ende der Verrohrung (Endposition \vec{P}_E , Endrichtung \vec{D}_E) anschließt und dieses damit vorgibt. Die Festlegung der Einbauposition im Packaging bestimmt damit die Endposition und Endrichtung der Verrohrung.

Eine mögliche Endposition und Richtung, aus der dargestellten (Teil-)Menge aller sinnvollen Endpositionen und Richtungen (rosa), ist in der Abbildung 6.5 grün dargestellt. Die linke Seite

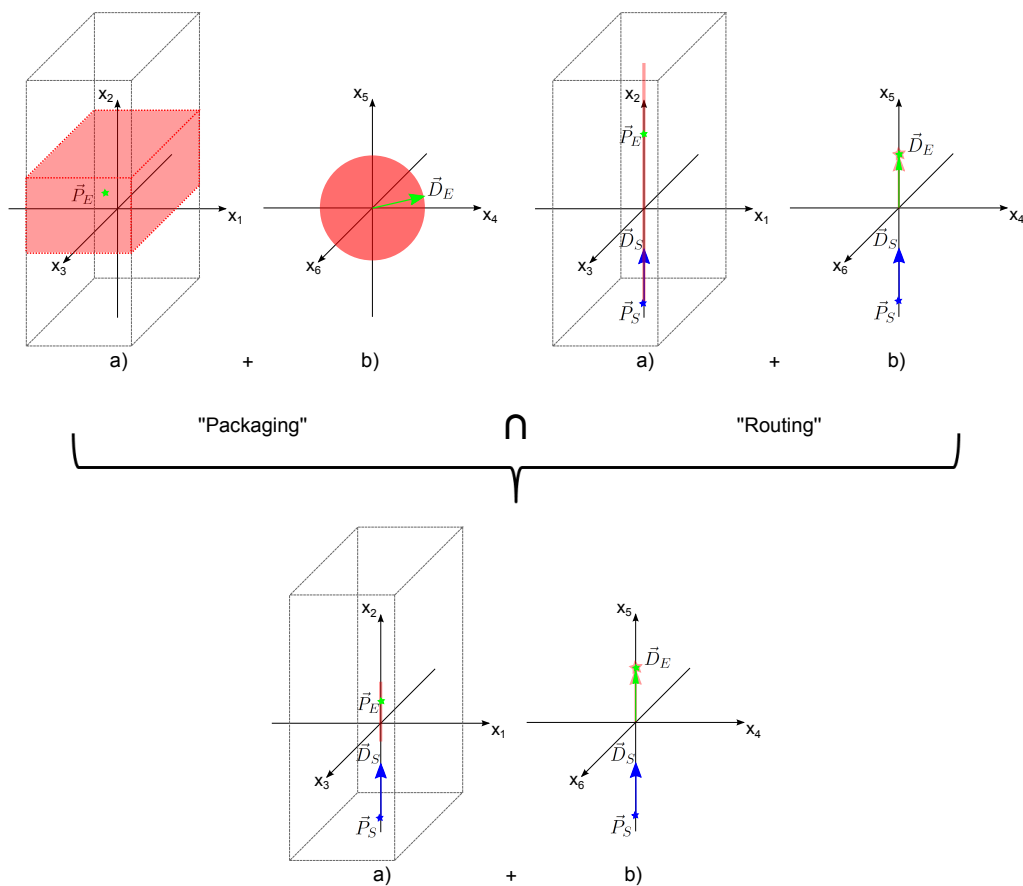


Abbildung 6.5: Mögliche Endpositionen \vec{P}_E (jeweils a)) und Richtungen \vec{D}_E (jeweils b)) als Teilmengen des Entwurfsraumes, bei fest vorgegebener Startposition \vec{P}_S und Richtung \vec{D}_S einer geraden Verrohrung, im Packaging und Routing einer Abgaslinie. Von oben nach unten: Synthese eines Entwurfes als Verschneiden der Teilmengen von Packaging und Routing im mathematischen Entwurfsraum.

a) einer Teilaufgabe stellt den jeweiligen Teil des Entwurfsraumes dar, der die Endpositionen repräsentiert. Analog stellt die rechte Seite b) den jeweiligen Teil des Entwurfsraumes dar, der die Endrichtungen wiedergibt.

Im Packaging (oben links a)) soll die Position der Komponenten, innerhalb des dargestellten rosa Quaders, frei variiert werden können. Gleiches gilt für die Ausrichtung der Komponente (oben links b)). Trotz der in a) und b) aufgesplitteten Darstellung des Entwurfsraumes darf nicht vergessen werden, dass es sich eigentlich um einen sechsdimensionalen Entwurfsraum handelt, der im betrachteten Beispiel *jeder* Einbauposition aus dem rosa Quader in a) jeweils alle in b) dargestellten Raumrichtungen zuordnet.

Im Routing (Abbildung 6.5 rechts) sind analog die Endpositionen und Endrichtungen der geraden Rohrstrecke für eine vorgegebene Startposition und Startrichtung dargestellt. Diese bilden auf Grund der Variablen Rohrlänge im Unterraum der möglichen Endpositionen der Verrohrungen (Abbildung 6.5 oben rechts a)) eine Linie. Im Unterraum der Endrichtungen der Verrohrung ergibt sich ein einzelner Vektor, der die fixierte Richtung des geraden Rohres repräsentiert (Abbildung 6.5 oben rechts b)).

Das Gesamtentwurfsproblem reduziert sich dann auf das Finden der Schnittmenge der beiden zu lösenden Entwurfsaufgaben beziehungsweise Teilsystemen Packaging und Routing³⁴. Diese Sichtweise des Entwurfes - das Verschneiden der, die zu kombinierenden Subsysteme repräsentierenden, Teilmengen des mathematischen Entwurfsraumes - deckt sich mit dem in Kapitel 2.2.6 dargestellten Constraints-Satisfaction-Problem, in dem durch die Hinzunahme von Zwangsbedingungen eine sukzessive Reduktion der Freiheitsgrade erreicht wird. Die Kombination zweier Zwangsbedingungen eines Constraints-Problem kann analog als das Verschneiden der jeweiligen, durch die isoliert betrachteten Zwangsbedingungen gebildeten, Teilmengen des mathematischen Entwurfsraumes aufgefasst werden.

In der Abbildung 6.5 ist das Ergebnis des Verschneidens der Teilmengen des Entwurfsraumes von Packaging und Routing unten dargestellt. Die resultierende Schnittmenge ist wiederum in rosa eingezeichnet. Als Resultat ergibt sich eine linienförmige (eindimensionale) Lösungsmenge bezüglich der möglichen Einbaupositionen der Komponente und eine feste Einbaurichtung (nulldimensionale Lösungsmenge), parallel zur Startrichtung der Verrohrung, durch die Beschränkung des Routings auf eine gerade Rohrstrecke.

Diese Vorgehensweise des Verschneidens entspricht einer parallelen Ausführung von Packaging und Routing und erlaubt im betrachteten Beispiel somit, im Gegensatz zur implementierten Entwurfssprache mit den sequenziellen Abfolgen von Packaging und Routing, den einfachen Entwurf mit gerader Rohrstrecke zu finden. Im Kontrast dazu ist die sequenzielle Abfolge dadurch gekennzeichnet, dass in der zuerst ausgeführten Entwurfsaufgabe (im Beispiel Packaging) eine Konfiguration festgelegt wird und die folgenden Entwurfsaufgaben mit dieser Teilkonfiguration, als fixierte Eingangsgröße, weiter verfahren.

Im Allgemeinen hängen aber die zu integrierenden und miteinander zu kombinierenden Subsysteme über zusätzliche Kopplungen voneinander ab und weisen mehr Freiheitsgrade (höherdimensionalere Räume) als das betrachtete Beispiel auf. Damit scheint es auf Grund der sich ergebenden hochdimensionalen mathematischen Entwurfsräume kaum möglich, in der gerade beschriebenen Vorgehensweise, alle Teilentwürfe/Subsysteme gleichzeitig auszulegen und zu integrieren. Um also nicht nur den Sonderfall entkoppelter Teilsysteme behandeln zu können, ist eine sinnvolle Sequentialisierung des Entwurfsprozesses notwendig.

6.1.4. Dimension der Schnittmenge

Für den Wert der Dimension der Schnittmenge zweier Teilmengen eines Vektorraumes lässt sich eine obere Schranke berechnen. Dazu soll zunächst von den Dimensionsformeln für das Verschneiden zweier nichtleerer, affiner Unterräume [29] A, B ausgegangen werden. Dabei wird eine nichtleere Schnittmenge von A und B vorausgesetzt (A und B nicht disjunkt):

$$\dim(A \cap B) = \dim(A) + \dim(B) - \dim(A \vee B). \quad (6.2)$$

Der Summand $\dim(A \vee B)$ stellt die Dimension des Verbindungsraumes $A \vee B$ beziehungsweise die affine Hülle von A und B dar. Dieser ist der kleinste affine Teilraum der A und B vollständig enthält. Die Dimension des Verbindungsraumes $\dim(A \vee B)$ lässt sich nach unten abschätzen:

$$\dim(A \vee B) \geq \max(\dim(A), \dim(B)). \quad (6.3)$$

³⁴Diese Sichtweise schließt aber nicht aus, dass es in den einzelnen Entwurfsaufgaben weitere Größen oder Variablen gibt, welche gegebenenfalls auch mit anderen Teilsystemen verschnitten werden müssen. In dem Beispiel wird nur der *gemeinsame* Teil des mathematischen Entwurfsraumes der beiden Teilsysteme, Packaging und Routing, betrachtet.

Setzt man Gleichung (6.3) in (6.2) ein, so erhält man folgende Ungleichung:

$$\dim(A \cap B) \leq \dim(A) + \dim(B) - \max(\dim(A), \dim(B)). \quad (6.4)$$

Dies lässt sich mit Hilfe der Beziehung

$$\dim(A) + \dim(B) = \min(\dim(A), \dim(B)) + \max(\dim(A), \dim(B)) \quad (6.5)$$

zu

$$\dim(A \cap B) \leq \min(\dim(A), \dim(B)) \quad (6.6)$$

umformen.

Diese obere Schranke (Gleichung (6.6)) für die Dimension der Schnittmenge zweier affiner Unterräume kann direkt auf Teilmengen dieser affiner Unterräume übertragen werden: Die Dimension der Schnittmenge kann durch die Beschränkung auf eine Teilmenge des affinen Unterraumes nicht größer werden, da durch die Beschränkung auf eine Teilmenge des affinen Unterraumes die Dimension des affinen Unterraumes, in den die Teilmenge eingebettet ist, nicht erhöht wird.

Die Dimension der affinen Unterräume und auch der Schnittmenge kann durch die Beschränkung auf Teilmengen nur gleich bleiben oder verringert werden, da es beim Verschneiden der Teilmengen auch noch auf die gegenseitige Lage dieser Teilmengen im Raum ankommt. Eine relative Verschiebung beziehungsweise ein nicht Aufeinanderliegen der Teilmengen führt nur zu einer Reduktion der Dimension der Schnittmenge.

Im oben dargestellten Ansatz stellen die Teilmengen des mathematischen Entwurfsraumes, als Repräsentation der Menge sinnvoller Entwürfe, Teilmengen affiner Unterräume dar. Prinzipiell lässt sich auch eine Abschätzung für fraktale Objekte, analog zur Ungleichung (6.6), berechnen. Die Fraktale, welche selbstähnliche Strukturen darstellen, sollen hier, auf Grund des in Abschnitt 2.5.3 dargestellten universellen Charakters des fraktalen Dimensionsbegriffes, ebenfalls betrachtet werden. Desweiteren kann nach [110] der Entwurf mit Entwurfssprachen selbstähnliche Strukturen auf verschiedenen Abstraktionsebenen hervorbringen.

Falconer gibt in [59] folgende Abschätzung für die Hausdorff-Dimension der Schnittmenge zweier Fraktale A und B (\dim_H sei die Hausdorff-Dimension von Fraktal X):

$$\dim_H(A \cap \sigma(B)) \leq \max\{0, \dim_H(A) + \dim_H(B) - n\} \quad (6.7)$$

Im Folgenden wird davon ausgegangen, dass die Fraktale nicht leer sind. Dabei ist n die Dimension des gemeinsamen Raumes, in dem die Fraktale eingebettet sind. σ steht für eine Transformation aus einer Transformationsgruppe beispielsweise einer Kongruenzabbildung³⁵ oder einer Ähnlichkeitstransformation³⁶. Die Gleichung (6.7) gilt nach [59] "praktisch" für alle Transformationen σ des Fraktals³⁷.

Die Tabellen 12 beziehungsweise 13 geben beispielhaft, für eine Dimension $n = 3$ eines die Objekte einbettenden Raumes, die oberen Schranken der Dimensionen der Schnittmengen zweier Untervektorräume beziehungsweise Fraktale A und B entsprechend den Gleichungen (6.6) und (6.7) für $\dim_{(H)}(A) \leq 3$ und $\dim_{(H)}(B) \leq 3$ an.

³⁵Lässt Winkel und Abstände invariant.

³⁶Lässt Winkel und Streckenverhältnisse invariant.

³⁷Es sind beispielsweise zwei Fraktale A, B mit $A = \sigma^o(B)$ denkbar, welche durch die spezielle Transformation σ^o in perfekte Überlagerung, $A \cap \sigma^o(B) = \sigma^o(B)$ gebracht werden können. Für eine solche Transformation σ^o würde Gleichung (6.7) nicht gelten.

Vektorraum: $\dim(A \cap B)$	$\dim(A) = 0$	$\dim(A) = 1$	$\dim(A) = 2$	$\dim(A) = 3$
$\dim(B) = 0$	0	0	0	0
$\dim(B) = 1$	0	1	1	1
$\dim(B) = 2$	0	1	2	2
$\dim(B) = 3$	0	1	2	3

Tabelle 12: Obere Schranken der Dimension der Schnittmenge zweier affiner Untervektorräume A und B nach Gleichung (6.6).

Fraktal: $\dim_H(A \cap B)$	$\dim_H(A) = 0$	$\dim_H(A) = 1$	$\dim_H(A) = 2$	$\dim_H(A) = 3$
$\dim_H(b) = 0$	0	0	0	0
$\dim_H(b) = 1$	0	0	0	1
$\dim_H(b) = 2$	0	0	1	2
$\dim_H(b) = 3$	0	1	2	3

Tabelle 13: Obere Schranken der Dimension der Schnittmenge zweier Fraktale A und B nach Gleichung (6.7) für eine Dimension $n = 3$ des einbettenden Raumes.

Es lässt sich im Folgenden zeigen, dass die für Fraktale gegebene, obere Schranke der Dimension der Schnittmenge in Gleichung (6.7) stets kleiner oder gleich ist als die in Gleichung (6.6) gegebene obere Schranke für affine Unterräume. Dazu werden zunächst die rechten Seiten der Gleichungen (6.6) und (6.7), unter Verwendung von $\dim_H = \dim$, zu der folgenden zu beweisenden Ungleichung zusammengesetzt:

$$\dim(A) + \dim(B) - n \leq \min(\dim(A), \dim(B)) \quad (6.8)$$

Ohne Beschränkung der Allgemeinheit soll $\dim(A) \leq \dim(B)$ angenommen werden. Nach Einsetzen auf der rechten Seite und Umformung ergibt sich:

$$\dim_H B \leq n. \quad (6.9)$$

Dies ist immer erfüllt, da n ja gerade die Dimension des Raumes ist, in dem A und B eingebettet sind. Für den Fall, dass die rechte Seite von (6.7) gleich Null ist, wird Gleichung (6.8) zu:

$$0 \leq \min(\dim(A), \dim(B)), \quad (6.10)$$

was ebenfalls immer erfüllt ist.

Dies bedeutet im Umkehrschluss, dass Gleichung (6.6) prinzipiell auch für Fraktale angewendet werden kann, auch wenn die mit der Ungleichung (6.7) bestimmte obere Schranke bessere (geringere) Werte für die obere Grenze der Dimension der Schnittmenge von Fraktalen liefert.

6.1.5. Dimension und Entwurfsablauf

Im vorherigen Abschnitt wurde dargestellt, dass die Dimension der Schnittmenge von den, die Entwürfe repräsentierenden, Teilmengen des mathematischen Entwurfsraumes entsprechend Gleichung (6.6) nach oben hin abgeschätzt werden kann. Dieses Ergebnis kann für die weitere Behandlung des Entwurfsprozesses folgendermaßen umformuliert werden:

Die niederdimensionalste Teilmenge des mathematischen Entwurfsraumes, welche jeweils (sinnvolle) Entwürfe von Sub- oder Teilsystemen repräsentieren, bestimmt die maximale Dimension

der Lösungsmenge (Schnittmenge) bei der Integration voneinander abhängiger Sub- oder Teilsysteme des Entwurfes.

Für die sequenzielle Abarbeitung des Entwurfsprozesses bedeutet dies, dass es prinzipiell vorteilhaft ist, erst die Subsysteme aus- und festzulegen, welche eine niederdimensionalere Teilmenge im gemeinsamen Unterraum des mathematischen Entwurfsraumes aufweist:

Bei sequenzieller Ausführung des Entwurfes und der Integration einzelner Teilsysteme sollte zuerst das Teilsystem, dessen Lösungsmenge (Teilmengen des mathematischen Entwurfsraumes) eine niedrigere Dimension im miteinander zu verschneidenden Unterraum des mathematischen Entwurfsraumes aufweist, bestimmt werden.

Diese Vorgehensweise setzt aber voraus, dass die Dimension der die möglichen Entwürfe repräsentierenden Teilmengen des mathematischen Entwurfsraumes bekannt sind. In Kapitel 6.1.3 wurde der Idealfall des simultanen Verschneidens der Teilmengen der zu integrierenden Teilsysteme dargestellt. Dieser setzt aber das Bekanntsein der jeweiligen Teilmengen und damit eine, zumindest im nichtdiskreten Entwurfsraum, potenziell unendliche Menge von (Teil-)Entwürfen voraus, da ein Entwurf nach den Darstellungen in Kapitel 6.1.1 prinzipiell empfindlich von den genauen Werten einzelner Variablen abhängen kann. Für explizit parametrisierte Teilsysteme des Entwurfes, wie beispielsweise die gerade Rohrstrecke des obigen Beispiels, lässt sich die Dimension direkt, aus dem beschreibenden Parametersatz, bestimmen.

Die Dimension der die Entwürfe repräsentierenden Teilmengen der zu integrierenden Systeme sollte aber einfacher zu bestimmen sein als diese exakte Form der Teilmengen, da die Dimensionen entsprechend Kapitel 2.5.3 prinzipiell aus den Skalierungseigenschaften der zu betrachteten Teilmengen bestimmt werden können. Dies kann unter Verwendung numerischer Methoden (zum Beispiel in [59]) näherungsweise geschehen, soll aber nicht mehr Teil dieser Arbeit sein, da zur Darstellung der Systematik die Herleitung ohne entsprechende numerische Berechnungen ausreichend ist. Die Systematik wird im Folgenden ausführlich begründet.

6.2. Begründung und Anwendung

Das oben dargestellte Prinzip der Bestimmung der dimensionsbasierten Entwurfsabfolge innerhalb eines sequenzialisierten Entwurfsszenarios soll in diesem Abschnitt anfangs logisch begründet werden und im weiteren Verlauf auf den Lösungspfadgenerator und das Zusammenspiel von Packaging und Routing explizit angewendet werden. Das Kapitel schließt mit einer kurzen, weiterführenden Betrachtung des Zusammenhanges von mathematischer Dimension der Teilsysteme des Entwurfes und der Bewertung der Komplexität des Entwurfes ab.

6.2.1. Mathematische Begründung der dimensionsabhängigen Entwurfsabfolge

Zunächst wird von zwei Teilsystemen A und B mit den Dimensionen $\dim(A) < \dim(B)$ ausgegangen. Entsprechend dem Prinzip der dimensionsabhängigen Entwurfsabfolge sollte die Abfolge mit der Auslegung von Teilsystem A beginnen und dann mit der Auslegung von Teilsystem B fortgesetzt werden.

Für die logische Begründung soll aber zunächst vom umgekehrten Fall ausgegangen werden. Dazu soll angenommen werden, dass zuerst Teilsystem B mit $\dim(B) = 2$ ausgelegt wird und die Dimension von Teilsystem A , $\dim(A) = 1$ beträgt. Als nächstes soll die Wahrscheinlichkeit $p(B \rightarrow A)$ betrachtet werden, dass bei Festlegung eines Entwurfes aus der zweidimensionalen

Menge der gültigen Entwürfe von Teilsystem B , dieser auch in der Menge der gültigen Entwürfe von Teilsystem A liegt und damit mit diesen kompatibel ist.

Dazu wird davon ausgegangen, dass bei sequentieller Ausführung der Auslegungen der Teilsysteme $B \rightarrow A$, damit in der Auslegung von B ohne Wissen über Teilsystem A , jeder mögliche Entwurf von Teilsystem B mit gleicher Wahrscheinlichkeit ausgewählt wird³⁸. Für die Bestimmung der Wahrscheinlichkeit soll ein Fall betrachtet werden, in dem die Teilmengen der gültigen Entwürfe in einem zweidimensionalen mathematischen Entwurfsraum eingebettet sind. Die Teilmenge der gültigen Entwürfe von Teilsystem B stellt dann ein flächiges Gebilde (rot) im mathematischen Entwurfsraum, siehe Abbildung 6.6, dar.

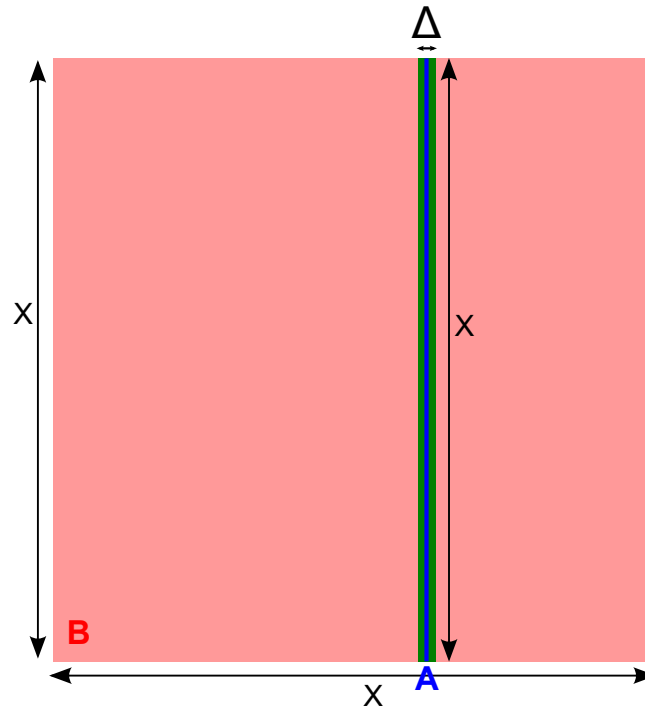


Abbildung 6.6: Teilmengen der gültigen Entwürfe der Teilsysteme A und B mit $\dim(A) = 1$ (blau) und $\dim(B) = 2$ (rot). Die eindimensionale Teilmenge von Teilsystem A ist eine Gerade. Diese soll aber eine Unschärfe Δ aufweisen können (grün).

Die eindimensionale Teilmenge des Teilsystems A (blau) liegt im "besten" Falle vollständig in der Ebene beziehungsweise in der Fläche der Teilmenge des Teilsystems B (rot). Für Teilsystem A soll eine Unschärfe Δ zugelassen werden (grün).

Damit kann die Wahrscheinlichkeit $p(B \rightarrow A)$ bestimmt werden. Diese ergibt sich zunächst als Quotient aus dem Flächeninhalt der grünen Fläche, im Grenzübergang zur eindimensionalen Linie mit verschwindender Unschärfe, und dem Inhalt der roten Fläche (Teilsystem B):

$$p(B \rightarrow A) \propto \frac{O(x \cdot \Delta)}{O(x^2)} \propto \frac{\Delta}{X}. \quad (6.11)$$

Die Länge X muss auf Grund der Annahmen $\dim(A) = 1$ und $\dim(B) = 2$ einen Wert größer Null haben. Für die Einhaltung von $\dim(A) = 1$ muss die Unschärfe Δ von Teilsystem A gegen Null gehen, so dass sich folgender Grenzwert der Wahrscheinlichkeit ergibt:

$$p(B \rightarrow A) \propto \lim_{\Delta \rightarrow 0} O\left(\frac{\Delta}{X}\right) \rightarrow 0. \quad (6.12)$$

³⁸Analog der Annahme des mikrokanonischen Ensembles der statistischen Physik [131].

Diese Betrachtung lässt sich auf andere Dimensionskombinationen $d_A = \dim A$ und $d_B = \dim B$ mit $d_A < d_B$ und $B \rightarrow A$ übertragen:

$$p(B \rightarrow A) \propto \lim_{\Delta \rightarrow 0} O\left(\frac{\Delta^{d_B-d_A}}{X^{d_B-d_A}}\right) \rightarrow 0. \quad (6.13)$$

Damit geht, bei einer sequenziellen Ausführung der Integration zweier Teilsysteme des Entwurfes, wenn mit der Auslegung des höherdimensionalen Systems ohne Vorwissen über das niederdimensionalere System begonnen wird, die Wahrscheinlichkeit einen gültigen Entwurf beziehungsweise eine Integration beider Teilsysteme zu finden, gegen Null. Dies konnte für den optimalen Fall, dass die niederdimensionalere Teilmenge in der höherdimensionalen Teilmenge liegt, gezeigt werden.

Die umgekehrte Reihenfolge, $p(A \rightarrow B) > 0$, kann ähnlich behandelt werden: Es soll zunächst wieder angenommen werden, dass eine gemeinsame Schnittmenge der zu verschneidenden Teilmengen der zu integrierenden Teilsysteme existiert. Dann ergibt sich eine Wahrscheinlichkeit $p(A \rightarrow B) > 0$ größer Null, in der (zufälligen) Auslegung des niederdimensionaleren Teilsystems A einen Entwurf zu finden, welcher auch im höherdimensionalen Teilsystem B liegt. Dies ist schematisch in Abbildung 6.7 dargestellt.

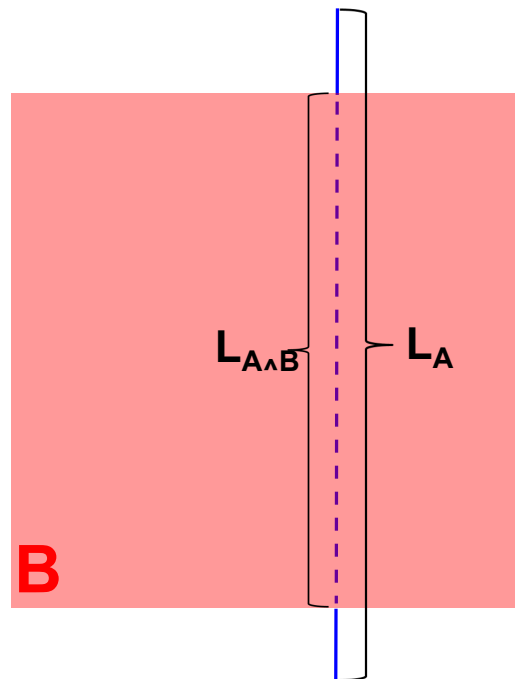


Abbildung 6.7: Schnittmenge der Teilmengen der gültigen Entwürfe der Teilsysteme A und B mit $\dim(A) = 1$ (blau) und $\dim(B) = 2$ (rot). Der Anteil der Länge $L_{A \wedge B}$ des gestrichelten Teils (Schnittmenge) der blauen Linie an der Gesamtlänge L_A der blauen Linie gibt die Wahrscheinlichkeit an, bei zufälliger Festlegung eines Entwurfes von Teilsystem A , eine gültige Konfiguration zu finden, welche auch in der Teilmenge der gültigen Entwürfe von Teilsystem B liegt.

Im betrachteten Beispiel von Abbildung 6.7 bestimmt der Quotient aus der Länge der Schnittmenge (blau gestrichelt) und der Gesamtlänge der Teilmenge von Teilsystem A (gesamte blaue Strecke) die Wahrscheinlichkeit $p(A \rightarrow B)$ einen gültigen Gesamtentwurf zu finden, wenn Teil-

systems A zufällig festgelegt wird. Der gültige Gesamtentwurf ergibt sich im Beispiel aus der erfolgreichen Integration der beiden Teilsysteme:

$$p(A \rightarrow B) \propto \frac{L_{A \wedge B}}{L_A} \geq 0. \quad (6.14)$$

Diese Betrachtung lässt sich ebenfalls auf andere Dimensionskombinationen $d_A = \dim A$ und $d_B = \dim B$ mit $d_A < d_B$ und $B \rightarrow A$ übertragen. Die Funktion $L^{\{x\}}$ soll das Maß der x -dimensionalen Teilmenge des Entwurfes definieren.

$$p(A \rightarrow B) \propto \frac{L_{A \wedge B}^{\{d_A\}}}{L_A^{\{d_A\}}} \geq 0 \quad (6.15)$$

Im entarteten Fall, wenn die Dimension $\dim(A \wedge B)$ der Schnittmenge $A \wedge B$ kleiner ist als die Dimension des niederdimensionaleren Teilsystemes A , gilt das Gleichheitszeichen in Gleichung (6.14) und (6.15) analog zur obigen Argumentation in den Gleichungen (6.12) und (6.13).

Im Allgemeinen liegt die Menge der gültigen Entwürfe eines Teilsystemes A (bildet keine Untermenge) nicht vollständig in der Menge der gültigen Entwürfe des anderen zu integrierenden Teilsystems B , $L_{A \wedge B}^{\{d_A\}} < L_A^{\{d_A\}}$, so dass gilt:

$$p(A \rightarrow B) \propto \frac{L_{A \wedge B}^{\{d_A\}}}{L_A^{\{d_A\}}} < 1 \quad (6.16)$$

Damit kann bei der sequentiellen Auslegung der Teilsysteme $A \rightarrow B$ ohne gegenseitiges Vorwissen der Teilsysteme das Auffinden einer Lösung nicht garantiert werden, da die Erfolgswahrscheinlichkeit in Gleichung (6.16) im Allgemeinen kleiner eins ist.

Die Misserfolgswahrscheinlichkeit, $1 - p(A \rightarrow B)$, gibt, unter den getroffenen Annahmen (unter anderem das Nichtvorhandensein von Auslegungswissen über das jeweils andere Teilsystem), die Häufigkeit von Iterationsschleifen. Solche Schleifen stellen eine Neuauslegung von Teilsystem A dar, wenn der ursprünglich ausgewählte Entwurf mit Teilsystem B inkompatibel ist.

In diesem Kapitel konnte damit gezeigt werden, dass bei der Integration zweier verschiedendimensionaler Teilsysteme des Entwurfes, bei sequenzieller Ausführung und unter Ausschluß von gegenseitigem Vorwissen der jeweils beteiligten Teilsysteme, die Wahrscheinlichkeit eine Lösung zu finden gegen Null geht, wenn mit der Auslegung des höherdimensionalen Teilsystems begonnen wird. Umgekehrt konnte gezeigt werden, dass die Wahrscheinlichkeit eine Lösung zu finden größer null ist, wenn mit dem niederdimensionaleren Teilsystem begonnen wird und eine gemeinsame Lösungsmenge existiert.

6.2.2. Dimensionsabhängige Entwurfsabfolge und Lösungspfad

Die vorgeschlagene dimensionsabhängige Entwurfsabfolge der Teilmengen A und B zweier zu integrierenden Teilsysteme mit $\dim(A) \leq \dim(B)$ kann direkt auf den in Kapitel 2.3.6 dargestellten Lösungspfadgenerator angewandt werden. Nach der vorgeschlagenen dimensionsabhängigen Entwurfsabfolge ergibt sich:

$$\text{Teilsystem } A \rightarrow \text{Teilsystem } B, \text{ da } d_A \leq d_B. \quad (6.17)$$

In der Lösungspfadbestimmung findet eine entsprechende Festlegung der Abfolge durch das Bestimmen bekannter Variablen statt. Die bekannten Variablen A' besitzen prinzipiell die Dimension $\dim(A') = 0$, da sie einen fixierten Wert besitzen. Die unbekannt Variablen B' , welche

im Gleichungssystem im Allgemeinen von den bekannten Variablen abhängen, können aber im Moment der Festlegung der bekannten Variablen noch alle Werte (beziehungsweise Werte eines vorgegebenen Intervalles) annehmen und haben damit eine höhere Dimension, $\dim(B') = 1$, als die bekannten Variablen. Erstere können in der Lösungspfadfindung des Lösungspfadgenerators auch erst nach Angabe der Werte der bekannten Variablen gelöst werden:

$$\text{Bekannte Variablen } \bar{A} \rightarrow \text{Unbekannte Variablen } \bar{B}, \text{ da } d_{\bar{A}} = 0 \leq d_{\bar{B}} = 1 \quad (6.18)$$

Der Vergleich der dargestellten Abfolgen (6.17) und (6.18), mit den entsprechend aufgetragenen Dimensionen der beteiligten Teilsysteme, macht deutlich, dass das Prinzip der dimensionsabhängigen Entwurfsabfolge auch auf die Lösungspfadfindung übertragen werden kann und dort Gültigkeit hat.

6.2.3. Anwendung der Dimensionsbetrachtung

Routing: Zielmenge Die Bestimmung der Dimensionen des oben (Abschnitt 6.1.3) betrachteten Beispiels, der Integration von Packaging und Routing, bedarf prinzipiell keiner numerischen Untersuchung. Die Dimension der geometrisch repräsentierten Teilmenge der zulässigen Einbaupositionen im Beispiel von Abbildung 6.5 ergibt direkt die Dimension $d_{Pos} = 3$, da es sich bei den zugelassenen Einbaupositionen um einen Würfel handelt. Die Dimension der zulässigen Einbaurichtungen ist ebenfalls $d_{Ric} = 3$, da alle Rotationsfreiheitsgrade im Einbau zulässig sein sollen. Dies ergibt, wie bereits erläutert, in dem betrachteten Beispiel eine Gesamtdimension für das Teilsystem Packaging von:

$$d_{Pack} = d_{Pos} + d_{Ric} = 6. \quad (6.19)$$

Das Routing einer geraden Rohrstrecke im betrachteten Beispiel von Abbildung 6.5 basiert auf einem fest vorgegebenen Startpunkt und einer vorgegebenen Startrichtung. Die Menge der Endpunkte bildet dann, wie oben bereits dargestellt, eine Linie und hat damit die Dimension $d'_{Pos} = 1$. Die Endrichtung ist durch die Startrichtung der geraden Verrohrung auf einen festen Wert vorgegeben und hat damit die Dimension eines Punktes $d'_{Ric} = 0$ (vergleiche Kapitel 6.1.3). Für das Teilsystem Routing ergibt sich dann eine Dimension von:

$$d_{Rout} = d'_{Pos} + d'_{Ric} = 1. \quad (6.20)$$

In dem Beispiel von Packaging und Routing ergibt sich eine dimensionsbasierte Entwurfsabfolge, in der zuerst das Routing und dann das Packaging der Komponenten ausgeführt werden sollte:

$$\text{Routing} \rightarrow \text{Packaging}, \text{ da } d_{Rout} < d_{Pack} \quad (6.21)$$

Dies bedeutet zuerst die Position und Länge des geraden Rohres im Bauraum zu bestimmen und dann, an der sich so ergebenden Endposition der Verrohrung, die sich der Verrohrung anschließende Komponente einzubauen. Anschließend kann im Packaging-Teilsystem diese Konfiguration darauf überprüft werden, ob die so bestimmte Einbauposition in der sechsdimensionalen Menge der zugelassenen Einbaupositionen liegt und so ein Schnittpunkt beider Teilmengen gefunden werden kann, welcher einen gültigen Entwurf von Packaging und Routing repräsentiert.

Im Falle einer Verrohrung, welche aus mehreren Rohrbögen besteht, nimmt die Zahl der Freiheitsgrade und damit die Dimension der möglichen Einbaupositionen und Richtungen zu. Entsprechend der Darstellung in Abbildung 4.19 von Kapitel 4.2.3 liegen die mit einem Rohrbogen

zu erreichenden Endpositionen auf einem Kegel. Die im Kegelmantel liegende und durch die Endposition der Verrohrung laufende Tangente des Kegels (Gerade zwischen Kegelspitze und Endposition), legt die Endrichtung des jeweiligen Bogens fest.

Betrachtet man die Dimension der Lösungsmenge für verschiedene Anzahlen von Rohrbögen, so kommen für jedes zusätzliche Rohrelement i , bestehend aus einem Rohrbogen mit angeschlossenen geraden Rohrstück variabler Länge, zwei zusätzliche Freiheitsgrade in das System der Verrohrung. Diese führen jeweils auf die kegelförmige Lösungsmenge eines Rohrbogens: Rotation θ_i des Rohrbogens um die Eintrittsachse und variable Länge L_i des zugehörigen geraden Rohrstückes (siehe auch Abbildung 4.19).

Führt man sich zunächst die mit einer gegebenen Anzahl von Rohrbögen erreichbaren Endpositionen und Richtungen vor Augen, so sind diese beiden Größen in einem sechsdimensionalen Unterraum des Entwurfsraumes aufgetragen. Diese Menge entspricht der in Kapitel 4.2.3 dargestellten Definition der Zielmenge Z der Verrohrung.

Die mathematische Dimension der Zielmenge kann aber nicht größer sein als $d = 6$, auch wenn durch Hinzufügen weiterer Rohrbögen die Zahl der Freiheitsgrade größer sein kann. Die abgeschätzte maximale Dimension d_{Ziel} der Zielmenge $Z(P_s, \vec{D}_s, I)$ skaliert damit folgendermaßen mit der Anzahl der Rohrbögen $|I| = N - 2$:

$$d_{Ziel} \leq \begin{cases} |I|, & |I| \leq 3, \\ 6, & |I| > 3. \end{cases} \quad (6.22)$$

Ein kurzes Gedankenexperiment soll veranschaulichen, dass die Dimensionalität dieser Mengen prinzipiell über den mathematischen Entwurfsraum variiert und damit ortsabhängig ist. Wählt man beispielsweise eine Verrohrung mit vier Rohrbögen, welche kleine Bogenwinkel $\bar{\alpha}_i < 22.5^\circ = \frac{90^\circ}{4}$ aufweisen, so kann prinzipiell nur eine Hälfte des Raumes der Zielpositionen erreicht werden: Der Halbraum, welcher ausgehend von der Startposition der Startrichtung entgegengesetzt ist, kann nur durch eine Bogenwinkelsumme von $\alpha_\Sigma > 90^\circ$ erreicht werden. Der entgegengesetzte Halbraum ist damit leer und hat eine Dimension von $dim(\emptyset) = -1$.

Diese Ortsabhängigkeit soll aber im Rahmen der Diskussion der Ergebnisse nicht weiter vertieft werden. Für weiterführende Arbeiten soll aber festgehalten werden, dass prinzipiell eine Ortsabhängigkeit der Dimension (vergleiche Multifraktale in Kapitel 2.5.3) vorliegen kann und damit eine entsprechende Multifraktalanalyse [178] für die Bestimmung der mathematischen Dimensionen der Teilmengen des mathematischen Entwurfsraumes angewandt werden muss.

Der umgekehrte Fall, $d_{Pack} \leq d_{Rout}$, liegt vor, wenn beispielsweise der Bauraum so knapp ist oder es andere Anforderungen gibt, so dass nur noch eine Einbauposition mit einer festen Richtung als sinnvoller Entwurf in Frage kommt. Dann würde sich für das Packaging eine Dimension von $\bar{d}_{Pack} = 0$ ergeben. Die Dimension der Zielmenge der Rohrstrecke würde entsprechend Gleichung (6.22) mit der Anzahl der Rohrbögen $|I|$ skalieren. Dies ergibt, nach der oben vorgeschlagenen dimensionsbasierten Entwurfsabfolge, folgende Entwurfsabfolge:

$$\text{Packaging} \rightarrow \text{Routing}, \text{ da } d_{Pack} \leq d_{Rout}. \quad (6.23)$$

Damit sollte die zu verrohrende Komponente zuerst im Bauraum platziert werden. Dies legt die Endrichtung und Endposition des nachfolgenden Routings fest. Im Teilsystem Routing muss dann eine mit dem Bauraum verträgliche Rohrstrecke, durch Variation der Rohrbogenkonfiguration I und Absuchen der jeweiligen Lösungsmengen $L(P_S, \vec{D}_S, P_E, \vec{D}_E, I)$ der Verrohrung, bestimmt werden.

Die untere Grenze der benötigten Freiheitsgrade beziehungsweise der Dimension des Routings lässt sich für Spezialfälle aus dem Vergleich der Start-/Endpositionen und Richtungen ableiten: Liegen beispielsweise die Start- und Endrichtungen windschief zueinander, so ist mindestens ein Rohrbogen mit der entsprechenden Anzahl von Freiheitsgraden notwendig. Für parallele Start- und Endrichtungen, deren zugeordnete Start- und Endpositionen aber nicht auf einer gemeinsamen Geraden liegen, kann direkt auf eine Mindestzahl von zwei Rohrbögen geschlossen werden (vergleiche auch das Routing der Rohrstrecken in [7]).

Routing: Lösungsmenge Die zusätzlichen Freiheitsgrade, welche für $|I| > 3$ über die in Gleichung (6.22) dargestellten Grenze hinausgehen, finden sich in der variablen Konfiguration der Zwischenpunkte beziehungsweise der Position und Ausrichtung der Rohrbögen im Inneren der Rohrstrecke wieder. Diese Freiheitsgrade bilden gerade die in Kapitel 4.2.3 definierte Lösungsmenge der Verrohrung.

Berücksichtigt man die Freiheitsgrade dieser Zwischenpunkte der Rohrstrecke in der vorliegenden Betrachtung, beispielsweise bezüglich der Prüfung ob die Rohrstrecke in den gegebenen Bauraum passt, so kommt man zu einer Erweiterung des (mathematischen) Entwurfsraumes.

Dazu soll zunächst eine vereinfachte Herangehensweise gewählt werden, welche nur die inneren Punkte \vec{P}_i mit $1 < i < N$ des die Rohrstrecke repräsentierenden Polygonzuges bezüglich der Einhaltung des Bauraumes berücksichtigt. Dazu soll für jeden der Punkte \vec{P}_i der Abstand zur Bauraumgrenze bestimmt werden, welcher größer sein muss als der halbe Rohrdurchmesser D :

$$\text{dist}_{\text{wand}}(\vec{P}_i) \geq \frac{D}{2}. \quad (6.24)$$

In der oben verfolgten Darstellung des Entwurfsproblems als Teilmenge des mathematischen Entwurfsraumes kann jeder dieser $|I|$ Zwischenpunkte \vec{P}_i in einem eigenen, im allgemeinen dreidimensionalen Unterraum aufgetragen werden. Die gültigen Entwürfe repräsentierende Teilmenge entspricht dann den Positionen \vec{P}_i , welche Gleichung (6.24) erfüllen. Die Positionen der Zwischenpunkte können nicht unabhängig voneinander variiert werden, da sie über die Zwangsbedingungen der Verrohrung gekoppelt sind. Insgesamt erhöht sich die Dimension des Entwurfsraums, bei einer Überprüfung der Bauraumeinhaltung für jeden der $|I|$ Zwischenpunkte, um:

$$d_{\text{zus}} = 3 \cdot |I|. \quad (6.25)$$

Im Allgemeinen Fall genügt es aber nicht, nur die Zwischenpunkte der Verrohrung bezüglich der Verträglichkeit mit den Bauraumgrenzen, zu überprüfen. Jeder Punkt \vec{P} der Verrohrung sollte auf seinen Abstand zu den Bauraumgrenzen überprüft werden. Dazu kann die Rohrstrecke mit $r \in [0, 1]$ parametrisiert werden:

$$\vec{P} = \vec{P}(r), \text{ mit } \vec{P}(0) = P_S \text{ und } \vec{P}(1) = P_E. \quad (6.26)$$

Im Allgemeinen, wenn beliebige Funktionen $P(r)$ als Verläufe der Rohrstrecke berücksichtigt werden sollen, ergibt sich folgende Bedingung für die Einhaltung der Bauraumgrenzen:

$$\text{dist}_{\text{wand}}(\vec{P}(r)) \geq \frac{D}{2} \text{ für } r \in [0, 1]. \quad (6.27)$$

Der Raum aller Funktionen $P(r)$ hat im Allgemeinen eine unendliche Dimension $\dim(P(r)) \rightarrow \infty$. Die Teilmenge der gültigen Entwürfe des unendlichdimensionalen Funktionenraumes [6] wird dann durch die Funktionen $\vec{P}(r)$ gebildet, welche Gleichung (6.27) erfüllen.

Die Suche nach den gültigen Entwürfen stellt sich im unendlichdimensionalen Funktionenraum in der Praxis als schwierig dar, da zusätzlich die Einhaltung der Verrohrungszwangsbedingungen weiterhin garantiert werden muss. Aus diesem Grund kann es in der praktischen Umsetzung durchaus Sinn machen, den Pfad $\vec{P}(r)$ in einem problemangepassten Raum mit reduzierter, endlicher Dimensionalität zu repräsentieren. Damit kommt man wieder auf die Repräsentation der Rohrstrecke als Polygonzug, mit ihrer endlichdimensionalen Basis [130], zurück.

Die beispielhaft durchgeführte Dimensionsbetrachtung zeigt damit als Nebenaspekt, dass gerade die gewählte Repräsentation der im Entwurf zu integrierenden Systeme und Objekte die Dimensionalität des Entwurfsproblems und damit die in der formalen Beschreibung des Entwurfs zu behandelnde Komplexität maßgeblich bestimmt. Die Forderung nach einer problemangepassten und minimalen Repräsentation (= Beschreibung) des Entwurfes, zur Minimierung der Komplexität, steht damit im Einklang mit der in Kapitel 2.2.4 nach Rudolph [154] dargestellten Forderung der Entwurfsbewertung nach einer minimalen Repräsentation. Diese minimale Repräsentation stellt nach Rudolph eine Voraussetzung für die Rückführung des Bewertungsproblems auf ein Beschreibungsproblem dar.

6.2.4. Einordnung der Ergebnisse

Die in diesem Kapitel betrachtete dimensionsbasierte Bestimmung eines Entwurfsablaufes kann in der Implementierung von Entwurfssprachen oder anderer Systeme zur Automatisierung des Entwurfsprozesses verwendet werden. Wie an den Ergebnissen der in dieser Arbeit implementierten Entwurfssprache zu erkennen ist, kann die Qualität der mit einer Entwurfssprache generierten Entwürfe, bei sequenzieller Ausführung von Entwurfsaufgaben, auch direkt von der Reihenfolge der Abarbeitung der Teilaufgaben im Entwurfsprozess abhängen.

Die in dieser Arbeit implementierte Entwurfssprache (Kapitel 3.1) weist eine sequenzielle Ausführung der Teilaufgaben auf, die in einer statischen bzw. festen Reihenfolge abgearbeitet werden. Dazu wurde in der vorliegenden Arbeit vor allem die statische Abfolge der Bestimmung der Einbaupositionen und der Verrohrung (*Routing* \rightarrow *Packaging*) betrachtet.

Die Ergebnisse (Kapitel 3.2 und 6.1.3) haben gezeigt, dass die fest implementierte Reihenfolge in der Bestimmung der Einbauposition der Komponenten praktisch nicht in der Lage ist, die Einbaupositionen so zu bestimmen, dass eine möglichst einfache Verrohrung bestimmt werden kann. Aus dieser Erkenntnis konnte der in Abschnitt 6.1.5 dargestellte dimensionsabhängige Entwurfsablauf gewonnen werden.

Die in Abschnitt 6.1.5 dargestellte implizite Verbindung der mathematischen Dimensionen mit der Abfolge der zu integrierenden Teilentwürfen kann in der Implementierung von Entwurfssystemen als Leitregel dienen. So sollte in der Implementierung des Produktionssystems eines sequenzialisierten Entwurfes die Möglichkeit vorgesehen werden, die Reihenfolge der Abarbeitung von miteinander gekoppelten Entwurfsaufgaben zur Laufzeit der Entwurfssynthese im Produktionssystem zu modifizieren.

Beispielsweise könnte die Entwurfsabfolge, auf Basis einer während der Laufzeit der Entwurfssynthese durchgeführten Abschätzung der mathematischen Dimensionen beteiligter Teilentwürfe, jeweils automatisiert neu festgelegt werden. Die Auslegung der Teilsysteme mit niedrigerdimensionaleren Teilmengen im gemeinsamen Entwurfsraum sollte dabei nach Abschnitt 6.1.5 vor der Auslegung der höherdimensionaleren Teilsysteme des gemeinsamen Entwurfsraumes erfolgen.

Eine solche automatisierte Auflösung der Entwurfssequenzen auf Basis der Dimensionalitäten der beteiligten Teilsysteme wäre ein möglicher Ansatz für die Implementierung eines deklarativen Produktionssystems. Ein solches Produktionssystem würde eine Analogie zum deklarativen Lösungspfadgenerator auf der Ebene des Produktionssystems darstellen und könnte so ein erster Schritt für die deklarative Behandlung des gesamten Entwurfsprozesses darstellen. Im idealisierten deklarativen Entwurfsprozess wäre es möglich Entwürfe nur auf Basis deklarativ gegebener Anforderungen zu erzeugen ohne eine Lösungssequenz definieren zu müssen.

Bisher implementierte Entwurfssprachen, beispielsweise im Luftschiff- oder Satellitene Entwurf [159, 165, 72, 197], werden im Wesentlichen zum konzeptionellen Vorentwurf von Systemen verwendet. Der konzeptionelle Vorentwurf ist im Allgemeinen durch einen recht groben Detaillierungsgrad sowie einen noch nicht im Detail optimalen Entwurf gekennzeichnet. In derlei Entwurfsszenarien kommt es nicht auf das Auffinden *des* optimalen Entwurfes an, sondern vielmehr um die Bewertung verschiedener Realisierungskonzepte eines Produktentwurfes. In diesen Fällen sollte eine, dem in dieser Arbeit dargestellten Zusammenhang von Dimension und Entwurfsabfolge widersprechende Entwurfsabfolge wegen der immanenten Unschärfe im Vorentwurf nicht ins Gewicht fallen.

Dies bedeutet aber, dass für zukünftige und fortgeschrittene Anwendungen der Entwurfssprachen, welche die Synthese detaillierter Entwürfe inklusive einer Feinauslegung des zu entwerfenden Produktes erfordern, die Abfolge der Entwurfshandlungen einer genaueren Betrachtung und Berücksichtigung in der Automatisierung des Entwurfsprozesses bedarf.

6.2.5. Weiterführende Betrachtung: Komplexität und Dimension im Entwurf

Nach Abbildung 6.2 gehen gerade die (Teil-)Systeme mit einer im Vergleich zum einbettenden mathematischen Entwurfsraum niedrigen Dimension (niedrige Zahl von Freiheitsgraden) eher mit einer hohen Zahl an Zwangs- und Randbedingungen und damit einer hohen Stufe der Sprachbeschreibung einher. Diese, durch die Einschränkung der Freiheitsgrade und durch die Berücksichtigung einer großen Zahl von Zwangsbedingungen spezialisierten Teilsysteme, weisen in der Regel eine höhere innere Komplexität auf.

Im Vergleich dazu weisen Teilsysteme, die nur wenige Zwangsbedingungen erfüllen müssen, eine höhere Zahl innerer Freiheitsgrade auf. Diese besitzen damit eine höhere mathematische Dimension, die sich der Dimension des einbettenden mathematischen Entwurfsraumes annähert.

Letztgenannte Teilsysteme, mit einer sehr hohen Zahl an Freiheitsgraden, stellen im Allgemeinen für den Entwerfer einfacher zu integrierende und damit weniger komplexe Systeme dar, da die hohe Zahl an Freiheitsgraden eine sehr flexible Integration der Systeme im Gesamtentwurf zulässt. Wird die Zahl der Freiheitsgrade und damit die Dimension der Menge der zugelassenen Entwürfe, im Vergleich zur Dimension des einbettenden mathematischen Entwurfsraumes, sehr weit eingeschränkt, so vereinfacht sich der Entwurf aus der Sicht des Entwerfers wiederum, da nur eine begrenzte Menge an Entwürfen betrachtet werden muss.

Dieses bivalente Verhalten der Komplexität C des Entwurfes (aus der Sicht des Entwerfers) in Abhängigkeit vom Verhältnis der Dimension der Teilmenge gültiger Entwürfe zur mathematischen Dimension des einbettenden Entwurfsraumes,

$$D = \frac{\dim(\text{gültige Entwürfe})}{\dim(\text{einbettender, mathematischer Entwurfsraum})}, \quad (6.28)$$

ähnelt dem typischen Verlauf der Komplexität eines Systems in Abhängigkeit der Entropie beziehungsweise der Unordnung U eines Systems im Bereich der Behandlung komplexer Systeme der statistischen Physik [45]. Ein solcher typischer Verlauf³⁹ ist in Abbildung 6.8 aufgetragen.

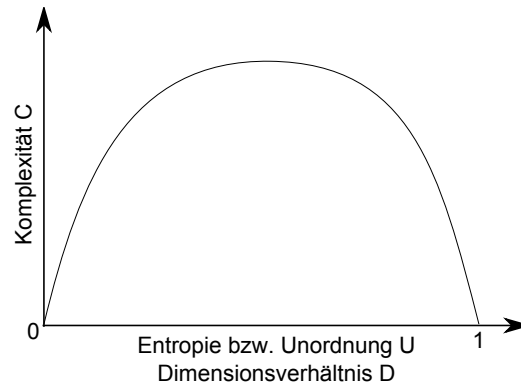


Abbildung 6.8: Prinzipielle Abhängigkeit der Komplexität eines Systemes von der Ordnung beziehungsweise der Entropie (nach [45, 173]). Ein analoger Verlauf lässt sich für das in Gleichung (6.28) definierte Dimensionsverhältnis D postulieren.

In der Darstellung von Abbildung 6.8 stellen Teilentwürfe mit einer hohen Zahl an Freiheitsgraden, $D \rightarrow 1$, Systeme geringer Komplexität bezüglich der Integration im Entwurf dar. Die große Zahl an Freiheitsgraden macht ein solches Teilsystem prinzipiell flexibel integrierbar und senkt so die Komplexität der Integration im Gesamtentwurf.

Teilentwürfe mit sehr wenigen Freiheitsgraden (wiederum im Vergleich zur mathematischen Dimension des einbettenden Entwurfsraumes), $D \rightarrow 0$, stellen in Abbildung 6.8 ebenfalls leicht zu integrierende Teilsysteme dar, da in diesem Grenzfall nur eine geringe Zahl an freien Parametern existiert, welche integriert werden müssen. Daraus ergibt sich eine geringer werdende Komplexität am unteren Ende der Skala des Dimensionsverhältnisses D in der Integration des Teilsystemes im Gesamtentwurf. Das Teilsystem ist, auf Grund der umfassenden Einschränkung der Freiheitsgrade, praktisch schon festgelegt und bedarf keiner komplexen Auslegung.

Die in diesem Abschnitt erfolgte, knappe Darstellung des Zusammenhanges der Komplexität und der mathematischen Dimension eines Entwurfes soll am Ende dieses Kapitels einen Hinweis auf weiterführende Untersuchungen und interessante zukünftige Aufgabenstellungen geben. Methoden und Konzepte aus dem Gebiet der Analyse komplexer Systeme könnten gerade im Bereich der Automatisierung des Entwurfsprozesses mit Entwurfssprachen neue Einblicke und Ansätze zur generischen Strukturierung des Entwurfsprozesses liefern.

³⁹Das archetypische Beispiel für den in Abbildung 6.8 dargestellten Verlauf der Komplexität ist der Wurf einer Münze: Eine perfekte Münze liefert *Kopf* und *Zahl* mit gleicher Wahrscheinlichkeit. Damit ist die Entropie beziehungsweise die Unordnung dieses Systems maximal. Eine solche Münze, mit jeweils zufälligem Ausgang, stellt aber kein sehr komplexes System dar, da es sich statistisch vorhersagen lässt und sonst keine weiteren relevanten Korrelationen aufweist. Eine gezinkte Münze soll immer das gleiche Ergebnis liefern und repräsentiert damit einen geordneten Zustand ohne Zufall, mit minimaler Entropie, und stellt ein System dar, dessen Verhalten ohne Schwierigkeiten vorhergesagt werden kann und damit nicht als komplex einzustufen ist. Erst eine Münze, mit einem Verhalten das zwischen den beiden geschilderten Extremen von idealem Zufall und idealer Ordnung liegt, kann als System mit einem komplexen Verhalten bezeichnet werden.

Teil IV.

Rückblick

7. Diskussion

Die folgende Diskussion soll verschiedene Aspekte (inhaltliche, implementatorische und anwendungsbezogene) der vorherigen Kapitel abschließend beleuchten. Dabei sollen vor allem weiterführende Fragestellungen und Anknüpfungspunkte betrachtet werden und die in der praktischen Implementierung, der in dieser Arbeit verwendeten Algorithmen und Entwurfssprachen gewonnenen Erfahrungen, dargestellt werden.

7.1. Inhaltliche Aspekte

Die dimensionsbasierte Entwurfsabfolge des letzten Abschnittes stellt für den Ersteller von Entwurfssprachen ein nützliches Instrument dar. Bei der Implementierung eines Produktionssystems in einer Entwurfssprache, das eine sequenzielle Abarbeitung von einzelnen Entwurfsaufgaben vorsieht, kann, unter Beachtung der mathematischen Dimensionen der am Gesamtentwurf beteiligten Teilentwürfe, schon im Voraus auf mögliche Komplikationen und Einschränkungen der jeweiligen Implementierung geschlossen werden.

Eine Umsetzung der dimensionsabhängigen Entwurfsabfolge könnte von der bloßen 'geistigen' (impliziten) Berücksichtigung bei der Implementierung einer Entwurfssprache durch den Entwerfer bis hin zur expliziten Berücksichtigung in einem Produktionssystem reichen. Erstere Umsetzung könnte helfen das versehentliche Ausschließen von Entwurfskonfigurationen, wie in dieser Arbeit beispielsweise in Abbildung 6.4 dargestellt, vorherzusehen und ggf. auszuschließen. Letztere Umsetzung könnte in einem deklarativen Produktionssystem mit einer automatisierten Bestimmung der mathematischen Dimensionen für die dynamische Festlegung einer Entwurfsabfolge zur Laufzeit der Entwurfssprache gipfeln.

Vom theoretischen Standpunkt wäre zunächst eine Dimensionsbestimmung verschiedener, praktisch ausgeführter (Teil-)Entwürfe sowohl analytisch als auch numerisch interessant. Damit könnte die praktische Umsetzbarkeit der dimensionsbasierten Entwurfsabfolge von Kapitel 6 überprüft werden. Ebenso könnte die Frage ob und in welchen Systemen es zu fraktalen Strukturen mit gebrochenen Dimensionen und Selbstähnlichkeit kommt, wie in [110] als generisches Entwurfsmuster postuliert, beantwortet werden.

Die in der mathematischen Begründung der dimensionsbasierten Entwurfsabfolge in Kapitel 6.2.1 vorgeschlagene, wahrscheinlichkeitsbasierte Abschätzung des Erfolges der Integration voneinander unabhängiger Teilentwürfe beziehungsweise Teilsysteme, mit einheitlichen Wahrscheinlichkeitswerten für alle berücksichtigten Entwürfe (mikrokanonische Gesamtheit), kann erweitert werden: Durch die Berücksichtigung von wechselseitigem Wissen, der zu integrierenden Teilsysteme, können die Wahrscheinlichkeitsverteilungen entsprechend angepasst werden.

Dabei könnten Wahrscheinlichkeitsdichten über die, die gültigen Entwürfe repräsentierenden, Teilmengen der mathematischen Entwurfsräume definiert werden, um im Voraus bekanntes Wissen über eine Entwurfsdomäne zu integrieren. Der Forschungsbericht [120] gibt einen Überblick über solcherlei stochastische Methoden zur Behandlung von Unsicherheiten im maschinellen

Lernen im Bereich von Expertensystemen, beispielsweise über Bayessches Netz und andere Werkzeuge. Ein solches maschinelles Lernen könnte in das Produktionssystem von Entwurfssprachen in die Bewertungs- und Rückführungsschleifen integriert werden (Abbildung 2.5).

Ein statistischer Ansatz könnte damit die Möglichkeit bieten a priori Wissen im nichtlinearen Entwurf zu berücksichtigen und somit die Wahrscheinlichkeit zu erhöhen, einen mit den Randbedingungen und Anforderungen verträglichen Entwurf zu synthetisieren, auch wenn durch die prinzipiell empfindliche Abhängigkeit von Anfangsbedingungen - wie in Kapitel 6.1.1 in der Betrachtung des Entwurfes als nichtlineares System festgestellt - jede betrachtete Entwurfsänderung ins Detail generiert und ausiteriert werden muss.

Die Berücksichtigung von Vorwissen in der Auslegung und Integration von Teilsystemen würde dem Entwerfen in der Praxis näher kommen: Bei der Integration von Teilsystemen des Entwurfes werden im Allgemeinen gewisse gegenseitige Annahmen und Vorwissen über sequenziell zu integrierende Teilsysteme berücksichtigt. Die in Kapitel 6 dargestellte, dimensionsabhängige Entwurfsabfolge ist formal auf den Grenzfall unabhängiger, zu integrierender Teilsysteme beschränkt. Aber auch dieser Grenzfall hat, wie in den Abschnitt 6.1.3 für die Abfolge des Packaging und Routing dargestellt, praktische Relevanz. Eine allgemeingültige Übertragbarkeit und Anwendbarkeit des Prinzips in Entwurfsanwendungen, deren zu integrierende Teilsysteme nicht unabhängig sind und über gegenseitiges Vorwissen verfügen, ist eine interessante zukünftige Fragestellung.

Der in Abschnitt 6.2.4 angedeutete Zusammenhang zwischen Komplexität eines Entwurfes und dem vorgeschlagenen Ordnungsmaß, das Verhältnis der Dimension der Teilmenge gültiger Entwürfe zur mathematischen Dimension des einbettenden Entwurfsraumes, kann in weiteren Arbeiten detailliert untersucht und überprüft werden. Die Komplexität [45] stellt ein wichtiges formales Maß in der Behandlung komplexer Systeme und den Ausgangspunkt für die Bestimmung des Vorhandenseins von Selbstorganisation [174] in einem System dar.

7.2. Implementatorische Aspekte

Bei der Implementierung der Plug-ins des Entwurfscompilers, für die Bedienung der Analyseprogramme in den Prozessketten, hat sich in der Umsetzung der Arbeit ein Gegensatz einer möglichst engen Integration der Analyseprogramme in den Entwurfscompiler und einer starken Kapselung der Analyseprogramme ergeben. Eine direkte und enge Integration der Analyseprogramme in den Entwurfscompiler, welche durch die Verwendung von OpenSource-Analyseprogramme wie OpenFOAM [137] oder OpenCascade [136] erst möglich wird, steigert zwar die Effizienz der Programmausführung, führt aber dazu, dass eventuelle Programmierfehler der zu integrierenden Programme, durch die enge Integration, auch die Stabilität des Entwurfscompilers und damit die Stabilität der Ausführung der Entwurfssprache gefährden.

Gerade für die Implementierung der Verrohrungssprache mit dem Optimierungsschema der Simulierten Abkühlung hat sich die starke Kapselung der auszuführenden Algorithmen und Funktionen (Starrkörpersimulator ODE [177], KKF-Konstruktion in OpenCascade [136]) in eigene, freistehende Programme als nicht optimal im Sinne der Laufzeiten dargestellt. Die Auslagerung in externe Programme führte aber auf eine extrem stabile Ausführung der Optimierungsalgorithmen, da Abstürze in den extern aufgerufenen Programmmodulen im Entwurfscompiler ohne Abbruch des Optimierungsprozesses abgefangen werden konnten. Auf diese Weise konnte dieser über mehrere Tage und Wochen autonom ausgeführt werden.

Die Anbindung des in C++ implementierten CAD-Kernels OpenCascade [136] über JNI (Java Native Interface) hat sich als problematisch erwiesen, da die in OpenCascade originär implementierte Speicherverwaltung mit der Java-eigenen Speicherbereinigung (Garbage Collection) in mehreren Fällen nichtdeterministisch kollidiert ist und so einer softwaretechnisch tiefergehenden Analyse bedarf.

Zur Steigerung der Effizienz der Analyseprogramme und der Berechnungsalgorithmen ist prinzipiell eine Parallelisierung dieser Softwareteile angezeigt, da der Trend in der Optimierung und Leistungssteigerung der Hardware in den letzten Jahren nicht mehr hauptsächlich über eine Steigerung der Taktfrequenz der Prozessoren, sondern über die Verwendung mehrerer Rechenkerne oder die Mitbenutzung des Graphikprozessors realisiert wird.

Mit steigender Rechenleistung können die Bauraumgrenzen direkt im Starrkörpersimulator der Verrohrungssprache berücksichtigt werden. Darüber hinaus könnte dann das Packaging ebenfalls im Starrkörpersimulator, parallel mit dem Routing, direkt im Konfigurationsraum erfolgen. Dies entspricht einer simultanen Bearbeitung beider Teilsysteme und macht somit Iterationschleifen, die durch die sequenzielle Abbarbeitung notwendig werden können, überflüssig.

7.3. Industrieller Einsatz

Die Abbildung 7.1 zeigt am Beispiel der Baumaschine aus Kapitel 3.2.1 noch einmal eine Übersicht der einzelnen Schritte der implementierten Entwurfssprache inklusive der ungefähren Laufzeiten der Einzelschritte. Die Automatisierung und daraus folgende Beschleunigung der Entwurfshandlung eines Produktes ist im industriellen Einsatz sicherlich der bedeutendste Treiber für den Einsatz einer computergestützten Entwurfsmethodik.

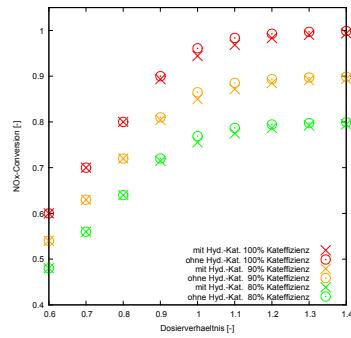
Die dargestellte Entwurfssprache und ihre Einzelkomponenten konnten schon im Laufe der Erstellung der Arbeit erfolgreich in verschiedenen industriellen Anwendungsprojekten eingesetzt werden. Der Einsatz von Entwurfssprachen ermöglichte im Projekteinsatz eine so schnelle Erzeugung der (CFD-)Berechnungsdaten und Modelle, dass nicht mehr die Modellerstellung der limitierende Faktor für die ausführbare Zahl an Berechnungen darstellte, sondern die zur Verfügung stehenden Berechnungskapazitäten der beschränkende Faktor war.

Für den breiten industriellen Einsatz der hier dargestellten Entwurfssprache sollte diese noch um die Erstellung komplexerer Abgasboxen und Abgasführungen ergänzt werden. Dies beinhaltet neben der Berücksichtigung von sich verzweigenden Strömungsführungen auch die Möglichkeit der Generierung von komplexen Abgasboxen mit integrierten Strömungsumlenkungen und verschachtelten Katalysatorträgern, wie sie in den Abgasanlagen moderner Nutzfahrzeuge eingesetzt werden. Dazu sollte die Entwurfssprache um die Modellierung von Freiformflächen erweitert werden. Für die Synthese von kleinstückzahligen SCR-Systemen ist aber die Verwendung, der in dieser Arbeit berücksichtigten, einfachen Standardelemente absolut ausreichend.

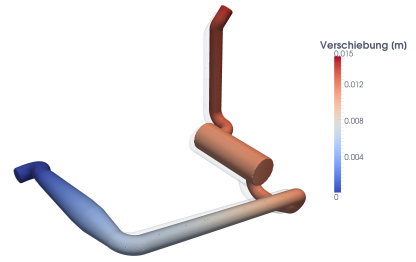
Für die breitere, industrielle Anwendung von Entwurfssprachen haben sich in zahlreichen Diskussionen mit größeren Unternehmen aus dem Bereich der Produktion und Entwicklung von Abgassystemen verschiedene Hindernisse herauskristallisiert: Der Einsatz der Entwurfssprachen erfordert eine starke Zusammenarbeit der Entwicklungsteams über Abteilungsgrenzen hinweg, da das Regelwerk zum Erstellen des domänenübergreifenden, universellen Datenmodells gemeinsam implementiert werden muss. Eine solche Zusammenarbeit scheint, auf Grund unternehmenspolitischer Gründe und zwischenmenschlicher Spannungen zwischen den beteiligten Akteuren, in der Praxis oftmals schwierig umzusetzen zu sein. Darüber hinaus erfordert die Implementierung einer Entwurfssprache breites Wissen im Bereich der Informationsverarbei-



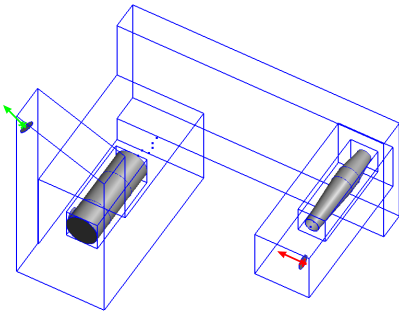
Analytische Auslegung
Δt ~ 1 min



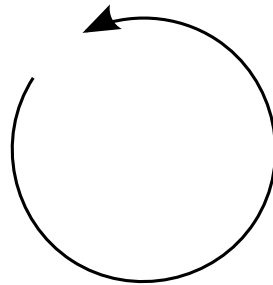
Post-Processing / Bewertung
Δt ~ 5 min



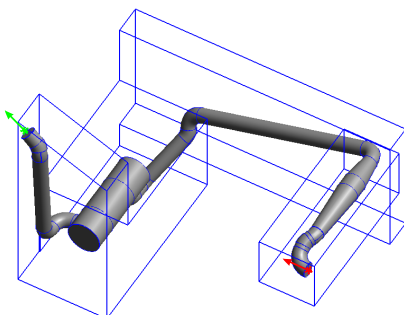
Strukturberechnung
Δt ~ 30 min



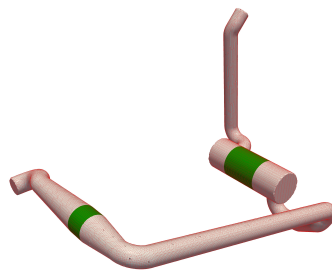
Packaging / Einbaupositionen
Δt ~ 5 min



CFD-Berechnung
Δt ~ 900 min



Routing / Verrohrung
Δt ~ 400 min



Vernetzung
Δt ~ 10 min

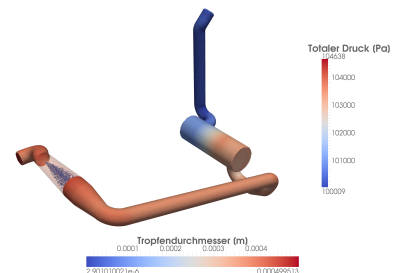


Abbildung 7.1: Übersicht und typische Laufzeiten des Beispiels Baumaschine der Entwurfssprache zum automatisierten Entwurf eines SCR-Systemes. Gesamtlaufzeit ca. 23h.

tung. Dies stellt eine nicht zu verachtende Schwelle für Ingenieure dar, in deren Studium im Allgemeinen kein besonderer Wert auf Informationsverarbeitung gelegt wird und wurde.

Sollen in den Prozess der integrierten Produktentwicklung mit Entwurfssprachen auch Zulieferunternehmen miteinbezogen werden, so muss die Möglichkeit bestehen, Teilentwurfssprachen auszutauschen, deren Know-How zu verdecken und zu verschlüsseln. So wird es möglich, dass die liefernden Unternehmen zwar die Modelle ihrer Produkte in Form einer Teilentwurfssprache, zur Integration in eine Gesamtentwurfssprache, zur Verfügung stellen, das Know-How ihrer Produkte in Form des detaillierten Regelwerks der entsprechenden Entwurfssprache aber durch eine adäquate Verschlüsselung verdeckt wird.

Mit steigenden Rechenleistungen und einer Parallelisierung, vor allem der CFD-Berechnungen, wird der Weg für die vollautomatisierte Exploration des Entwurfsraumes und die evolutionäre Optimierung von SCR-Systementwürfen, ohne manuelle Eingriffe und mit voller simulativer Validierung, frei. Zukünftige Rechenleistungen sollten den endgültigen Durchbruch des wissensbasierten Entwurfes (mit Entwurfssprachen) befördern und nach der Rationalisierung der Produktion während der vergangenen Jahrzehnte, auch auf eine Effizienzsteigerung des Entwurfes durch rechnergestützte Automation führen.

8. Zusammenfassung

Das primäre Ziel der Arbeit war die Erstellung einer graphenbasierten Entwurfssprache zur Automatisierung des Entwurfes von SCR-Abgassystemen. Diese führt auf eine erhebliche Beschleunigung des Entwurfsprozesses, sowohl in der Geometriegenerierung als auch in der Synthese und Auswertung der CFD-Modelle. Die Doppelseite mit den Abbildungen 8.1- 8.4 zeigt dazu eine kleine Auswahl von zweisträngigen Abgasanlagen in verschiedenen Bauräumen, welche mit der Entwurfssprache dieser Arbeit vollautomatisiert erzeugt wurden.

In der Modellierung dieser Entwurfssprache konnte durch die Verwendung des Entwurfscompiler 43V2 [90] auf ein fertig implementiertes Rahmenwerk zurückgegriffen werden. Dieses stellt die wichtigsten Funktionalitäten zur Erstellung einer Entwurfssprache (Lösungspfadgenerator, (graphische) Editoren zur Verwendung der Modellierungssprachen,...) zur Verfügung und besitzt einfache Mechanismen zur funktionalen Erweiterung, beispielsweise für die Implementierung von Plug-ins zur Kommunikation mit Analysesoftware.

Die Mächtigkeit dieses Ansatzes - der regelbasierter Aufbau eines zentralen Modelles in einer Modellierungssprache, die automatisierte Behandlung von Gleichungssystemen und Ausführung des Modelles in einem Entwurfscompiler - zeigt sich eindrucksvoll: Der Datenaustausch zwischen (Analyse-)Programmen wird nahtlos ermöglicht und gleichzeitig automatisiert. Ingenieurwissen wird ausführbar gemacht und kann wiederverwendet werden. Die im Rahmen dieser Arbeit entstandene Entwurfssprache konnten von Beginn an im produktiven, industriellen Einsatz in verschiedensten Anwendungsprojekten und Szenarien eingesetzt werden und hat so zu einer erheblichen Produktivitätssteigerung sowie Qualitätssicherung in der Auslegung und Berechnung von SCR-Abgasreinigungssystemen geführt.

Die Entwurfssprache zur automatisierten Synthese der CAD-Modelle von Verrohrungen in einem digital gegebenen Bauraum, unter Verwendung von Standardrohrbauteilen, baut auf der Verwendung eines Starrkörpersimulationsprogrammes [205] auf. Die Verrohrungssprache stellt prinzipiell eine Gestaltsgrammatik [8] dar, welche mit einem statistischen Optimierungsalgorithmus [103] gekoppelt ist. Diese Herangehensweise stellt ein alternatives Lösungskonzept, im

Vergleich zu den in der Automatisierung des Verrohrungsentwurfes allgemein eingesetzten gitterbasierten Methoden [7], dar.

Der im Entwurfscompiler integrierte Lösungspfadgenerator [171] wurde im Rahmen der Arbeit weiterentwickelt. Dazu wurde ein lokales Regelwerk identifiziert, welches auf eine selbstassemblierende Bestimmung der Lösungssequenz eines Zwangsbedingungsnetzwerkes führt, die sich generisch parallelisieren lässt und einen erheblichen Performancevorteil bringt.

Sowohl die Verrohrungssprache, mit der Verwendung eines statistischen Optimierungsalgorithmus, als auch der selbstassemblierende Lösungspfadgenerator setzen Lösungsprinzipien aus dem Bereich komplexer Systeme und der statistischen Physik ein.

Aufbauend auf diesen Implementierungen und Problemlösungsstrategien konnte mit Hilfe der Analyse der mathematischen Dimensionen der am Entwurf beteiligten und zu integrierenden Teilsysteme, welche ebenfalls ein Instrument der Charakterisierung komplexer Systeme darstellt, eine allgemeine Regel für die Abarbeitungsrichtung eines sequenzialisierten Entwurfsprozesses abgeleitet werden.

8.1. Ergebnisse

Die wichtigsten Ergebnisse der einzelnen Schwerpunkte der Arbeit sind in folgender Auflistung, mit einem entsprechenden Verweis auf das jeweilige Kapitel, dargestellt:

- **Gesamtentwurfssprache:** Die aus modularen Teilentwurfssprachen zusammengesetzte Entwurfssprache, zur Entwicklung von parallelen, einlinigen SCR-Systemen in einem gemeinsamen und digital vorgegebenen Bauraum, ermöglicht die im SCR-Systementwurf relevanten Modelle autonom zu erzeugen. Damit reduziert sich der kostenintensive Personaleinsatz in der manuellen Auslegung, Aufbereitung und Erstellung der beteiligten digitalen Modelle auf einen Bruchteil. Dies befreit den Ingenieur von wiederkehrenden Standardaufgaben und gibt Zeit für kreatives Arbeiten an neuen Lösungen. Die Überwachung der Ausführung der Entwurfssprache kann idealerweise 'nebenher' erfolgen.

Neben der Reduktion des Personalaufwandes durch die Automatisierung der Modellgenerierung und Aufbereitung wird durch den Einsatz graphenbasierter Entwurfssprachen auch eine Verkürzung der Gesamtdauer eines Entwurfes erreicht: Die zuvor manuell durchgeführten Arbeiten werden vollständig auf den Computer verlagert und die effektive Laufzeit der Validierung eines SCR-Systementwurfes schrumpft annähernd auf die reinen Laufzeiten der Generierung der Rohrstrecken und der Berechnung der Simulationsmodelle.

Das digitale Abbild des Entwurfsprozesses von SCR-Systemen in einer Entwurfssprache führt zu einer Konservierung und implizierten Standardisierung des Ingenieurwissens sowie auf die Möglichkeit der Wiederverwendung des entsprechenden Know-Hows im Rahmen eines computergestützten, virtuellen Entwurfsprozesses (\rightarrow Kapitel 3).

- **Verrohrungssprache:** Mit Hilfe der dargestellten KKF-Konstruktion ist es möglich, Regeln zur Manipulation einer auf Standardrohrbauteilen basierenden Rohrstrecke zu definieren. Diese Regeln erlauben das Austauschen, Entfernen und das Hinzufügen von Rohrbögen in einer gegebenen Rohrstrecke an einer vorgegebenen Position. Die Konsistenz der Rohrstrecke, bei Manipulation und Variation der Rohrbogenkonfiguration, wird durch die KKF-Konstruktion gewährleistet.

Die Simulation der Verschiebungs- und Rotationsfreiheitsgrade der Rohrstrecke in einem Starrkörpersimulator erlaubt eine parametrische Variation der Rohrstrecke im Konfigurationsraum unter Berücksichtigung der Ausdehnung der Rohrbögen sowie eine mögliche Berücksichtigung von Bauraumverletzungen und gegenseitigen Durchdringungen paralleler Rohrstrecken. Die Kopplung dieser Manipulationen mit einem Optimierungsverfahren (Simulierte Abkühlung) führt auf eine vollautomatische Synthese von Rohrstrecken in einem gegebenen Bauram, unter Verwendung von Standardrohrbauteilen und bei Berücksichtigung vorgegebener Start-, End- und sonstiger Randbedingungen (\rightarrow *Kapitel 4*).

- **Selbstassemblierender Lösungspfadgenerator:** Das graphenalgorithmische Regelwerk des Lösungspfadgenerators lässt sich in ein lokales, selbstassemblierendes Regelwerk überführen. Diese Regelwerk, das auf der Ebene der am Gleichungssystem beteiligten Gleichungen und Variablen definiert ist, bedarf keiner globalen Algorithmen, welche das Gleichungssystem im Ganzen erfassen, sortieren und manipulieren. Die Anwendung des selbstassemblierenden Lösungspfadgenerators setzt die Abwesenheit zyklischer Abhängigkeiten voraus, was in den bisher implementierten Entwurfssprachen aber praktisch immer der Fall war.

Der Übergang zum lokal formulierten Regelwerk erlaubt eine einfache Implementierung und generische Parallelisierung des Regelwerkes und führt auf eine erhebliche Reduktion der Laufzeiten im Vergleich zur graphenalgorithmischen Referenzimplementierung [171]. Je nach Verwendeter CPU Zahl (eine bis vier CPUs) ergibt sich eine Beschleunigung um den Faktor drei bis zehn (\rightarrow *Kapitel 5*).

- **Dimensionsabhängige Entwurfsabfolge:** In einem sequenzialisierten Entwurfsprozess führt die Bestimmung der mathematischen Dimensionen der Mengen, welche die gültigen Entwürfe zweier zu integrierender Teilsysteme repräsentieren, auf eine bevorzugte Abarbeitungsreihenfolge der Auslegung der beteiligten Teilsysteme. Der Entwurf des niederdimensionaleren Teilsystems, bezogen auf den gemeinsamen Entwurfsraum der zu integrierenden Systeme, sollte immer zuerst ausgeführt werden. Nur in dieser Entwurfsabfolge besteht eine endliche Wahrscheinlichkeit, dass der so festgelegte Teilentwurf eine kompatible Konfiguration im höherdimensionaleren Teilsystem besitzt, welche zu einem gültigen Gesamtentwurf zusammengesetzt werden kann.

Diese Erfolgswahrscheinlichkeit ist bei sequenzieller Ausführung, ohne gegenseitiges Auslegungswissen der ungekoppelten Teilsysteme, im Allgemeinen kleiner eins und erfordert so Iterationsschleifen im Entwurfsprozess. Die mathematische Dimension eines Teilsystems des Entwurfes kann ein (dimensionsloses) Maß der Komplexität des Systems darstellen (\rightarrow *Kapitel 6*).

8.2. Ausblick

Ein erster Anknüpfungspunkt sich anschließender Arbeiten stellt die in der Diskussion in Kapitel 7 herausgearbeiteten, noch bestehenden Einschränkungen und offenen Fragen der Entwurfssprache der vorliegenden Arbeit dar. Für den industriellen Einsatz hervorzuheben sind dabei primär eine weitere Erhöhung der geometrischen Komplexität der zu generierenden Systeme, im Zusammenspiel mit einer schnelleren Berechnungsinfrastruktur, welche geschlossene oder evolutionäre Optimierungsschleifen der Anlagenentwürfe ermöglicht.

Die Dimensionsbestimmung in Kapitel 6, welche den inhaltlichen Teil der Arbeit abschließt, verspricht aus der Sicht der Theorie komplexer Systeme das weiteste Betätigungsfeld für zu-

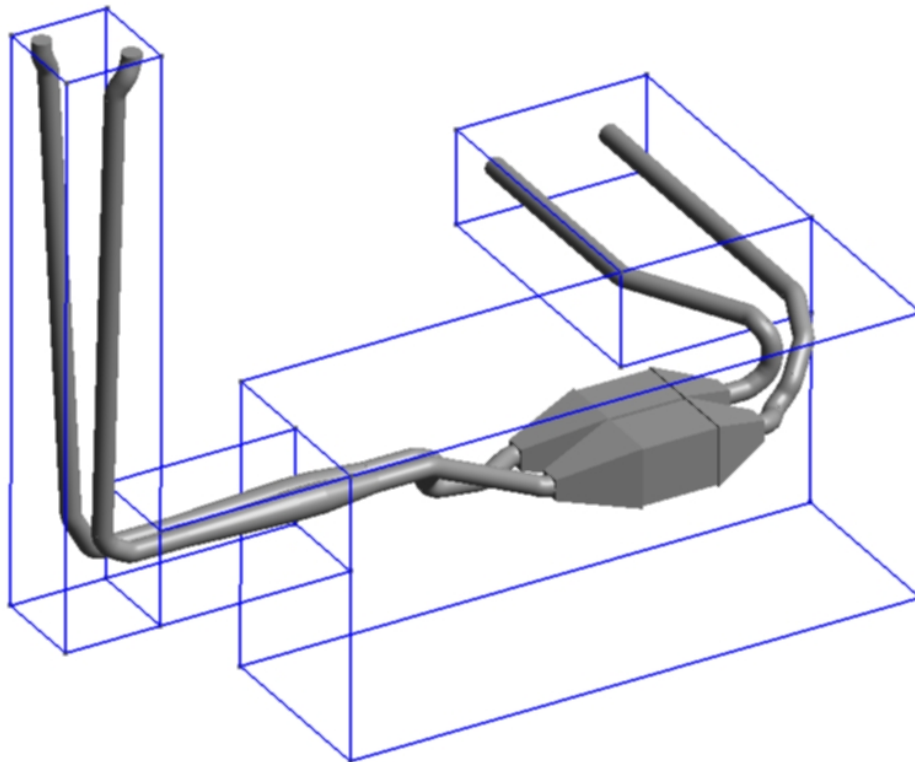


Abbildung 8.1: Zweisträngiges SCR-System mit quaderförmigen SCR-Katalysatoren und axialer An- und Abströmung im vorgegebenen Bauraum (Drahtgitter).

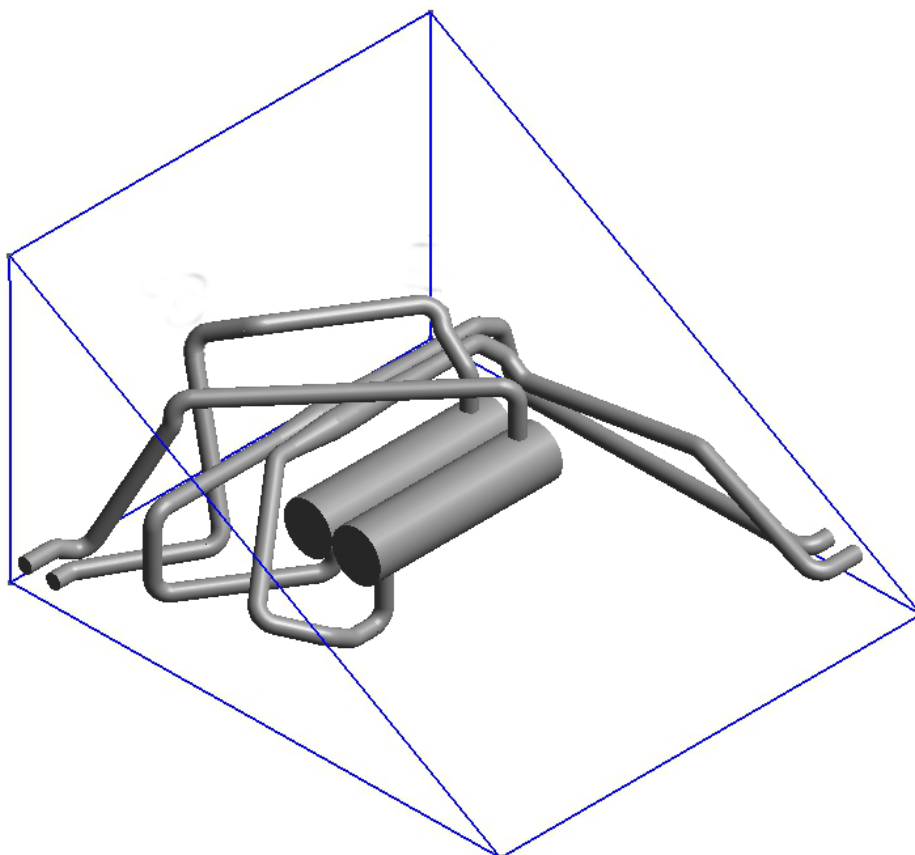


Abbildung 8.2: Zweisträngiges SCR-System mit zylindrischen SCR-Katalysatoren und radialer An- und Abströmung im vorgegebenen Bauraum (Drahtgitter).

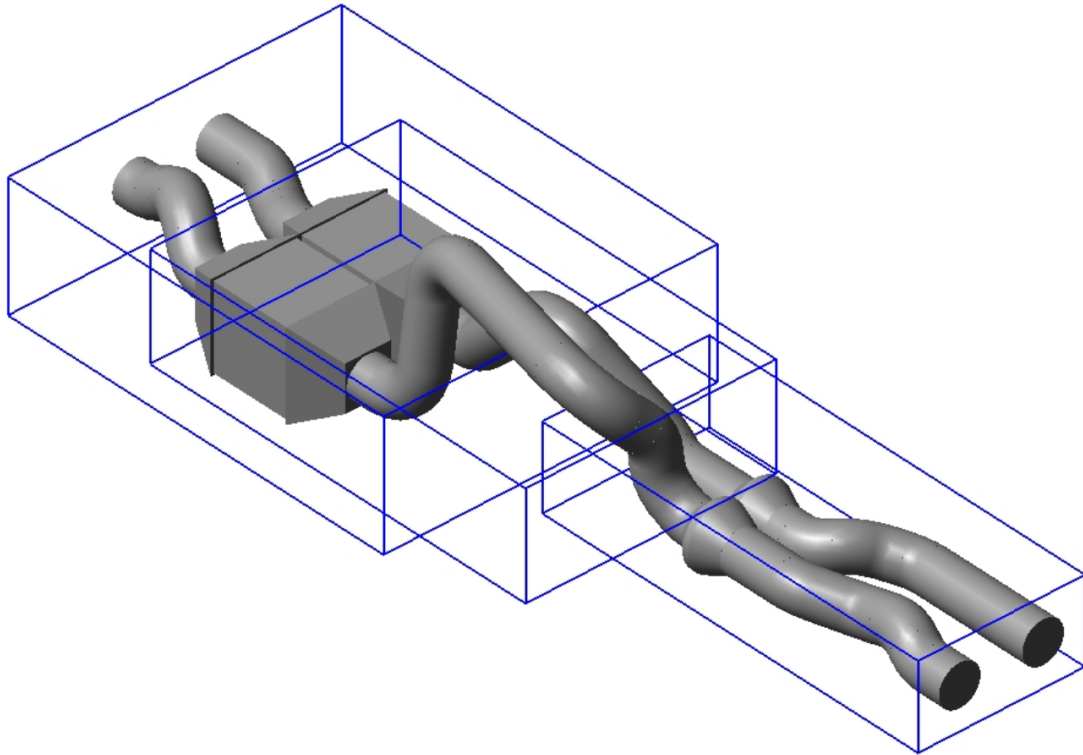


Abbildung 8.3: Zweisträngiges SCR-System mit quaderförmigen SCR-Katalysatoren und axialer An- und Abströmung im vorgegebenen Bauraum (Drahtgitter).

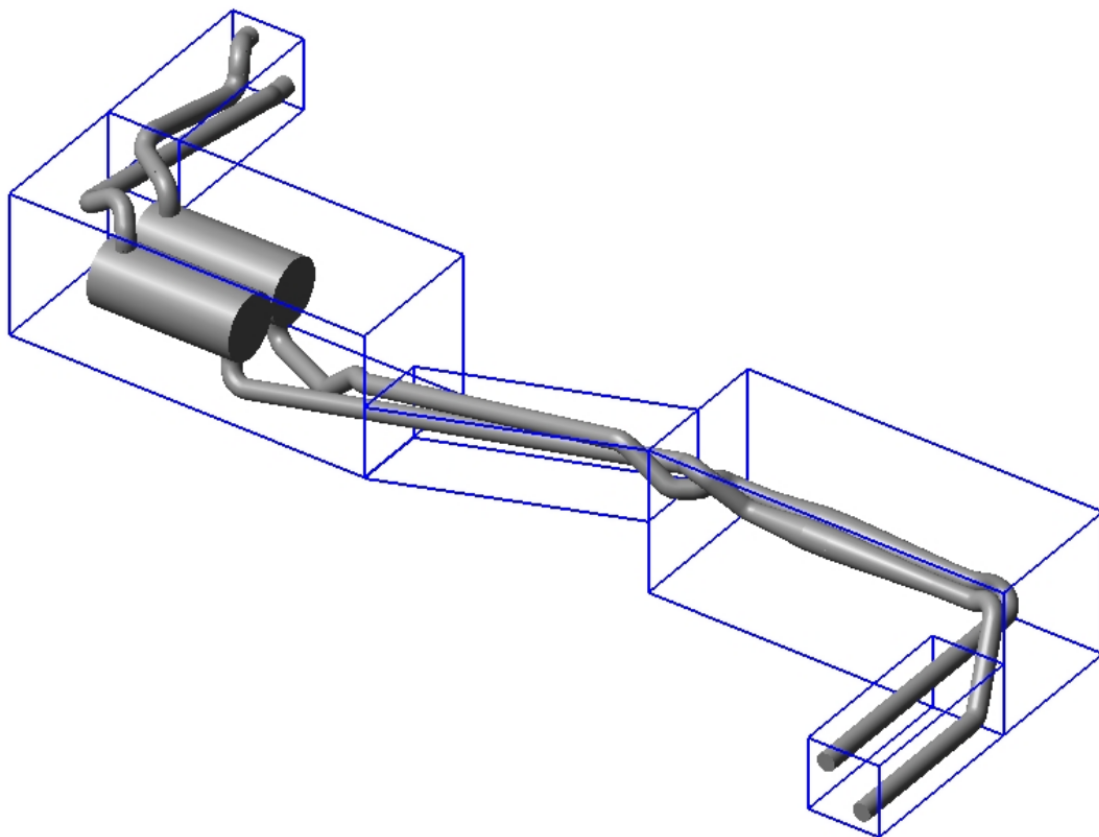


Abbildung 8.4: Zweisträngiges SCR-System mit zylindrischen SCR-Katalysatoren und radialer An- und Abströmung im vorgegebenen Bauraum (Drahtgitter).

künftige Arbeiten: Die systematische Betrachtung der Komplexität sowie des Zusammenspiels der Teilsysteme einer Entwurfssprache kann zukünftig helfen, (systemische) Risiken eines komplexen und automatisierten Entwurfes in einem frühen Stadium aufzudecken und gegebenenfalls formal zu quantifizieren.

Gerade im Hinblick auf die Robustheit und Fehlertoleranz eines Entwurfs(-Systemes) stellt die konsistente Quantifizierung der Systemkomplexität ein wichtiges Maß dar. So postuliert beispielsweise Andrew Zolli plakativ in einem Blog⁴⁰ (abgerufen am 19.04.2014), dass die Resilienz⁴¹ und die Komplexität eines Systems (prinzipiell) im Widerspruch stehen:

”Want to Build Resilience? Kill the Complexity!”

Der mit Hilfe von Entwurfssprachen formalisierte Entwurf komplexer Produkte stellt geradezu eine Einladung dar, solcherlei Aussagen und Fragestellungen zu untersuchen. Die in den letzten Jahren bekannt gewordenen technischen Schwierigkeiten in Großprojekten wie Stuttgart 21, der Neubau des Berliner Flughafens ”Willy Brandt” oder die Schwierigkeiten in der Entwicklung der Flugzeuge A350 und A380 von Airbus, beziehungsweise des ”Dreamliners” von Boeing unterstreichen den in der Durchführung komplexer Projekte immer wichtiger werdenden Aspekt der Fehlerrobustheit und Störungstoleranz.

Eine weitere Fragestellung des wissensbasierten Entwurfes stellt der zu erreichende maximale Grad der Automatisierung des Entwurfes dar: Gibt es prinzipielle Grenzen der Automatisierung? Kann das Prinzip der Selbstassemblierung, welcher im Lösungspfadgenerator umgesetzt wurde, auf den gesamten Entwurfsprozess übertragen werden? Wie könnte solch ein deklarativer Gesamtentwurfsprozess, ohne äußere Vorgabe der Regelabfolge und der Entscheidungen im Produktionssystem, aussehen? (Wie) Können sich intelligente Vokabeln oder informationstechnisch stark gekapselte Teilsysteme über lokal formulierte, generische Regeln, auf Basis gegenseitiger Interaktionen der Teilsysteme, unter Einhaltung deklarativ gegebener Anforderungen und Randbedingungen, zu einem gültigen Entwurf selbständig zusammensetzen?

Ein solcher selbstassemblierender Entwurf würde dann ausschließlich durch die Definition der Randbedingungen und Anforderungen deklarativ bestimmt. In den Entwurfssprachen könnten die kombinatorisch möglichen Zusammenbauregeln prinzipiell aus der Struktur des Klassendiagrammes und damit der gegenseitigen Beziehungen der Vokabeln bestimmt werden [156]. Die in Kapitel 6 definierte dimensionsabhängige Entwurfsabfolge könnte ein Baustein sein, der zur (automatisierten) Definition der Regelabfolge und damit zur Bestimmung des Produktionssystems eines solchen selbstassemblierenden Entwurfssystems dienen könnte.

Es kann an dieser Stelle nur spekuliert werden, ob die in Kapitel 6 dargestellte statistische Behandlung und Begleitung des Entwurfsprozesses zur Integration von a priori Wissen über die am Entwurf partizipierenden Teilsysteme verwendet werden kann. Eine solche Integration von a priori Wissen könnte ein möglicher Baustein zur Reduktion der sich aus dem Klassendiagramm ergebenden kombinatorischen Vielfalt sein.

Diese Auswahl an offenen und wissenschaftlich sicher herausfordernden Fragen zeigen, welches breite und sicher höchst interessante Forschungsfeld durch die Entwurfssprachen eröffnet wird. Die Entwurfssprachen bieten die Chance, durch die vollständige Abbildung und Formalisierung des Entwurfsprozesses, Methoden der Mathematik, Statistik und der komplexen Systeme im Bereich des Produktentwurfes interdisziplinär anzuwenden und so, durch eine Strukturierung

⁴⁰Harvard Business Review Blog Network:

[HTTP://BLOGS.HBR.ORG/CS/2012/09/WANT_TO_BUILD_RESILIENCE_KILL_THE_COMPLEXITY.HTML](http://blogs.hbr.org/cs/2012/09/want_to_build_resilience_kill_the_complexity.html)

⁴¹Resilienz beschreibt die Fähigkeit eines Systems, sein originäres Verhalten auch beim Auftreten einer Störung zu erhalten oder wiederzugewinnen.

und Automatisierung des Entwurfsprozesses, die Produktivität des Entwerfens um Größenordnungen zu steigern.

Die in dieser Arbeit vollzogene Anwendung und Übertragung der Entwurfssprachen auf den Entwurf von SCR-Systemen ist ein Indiz für die generische Natur dieses Automatisierungskonzeptes und lässt erwarten, dass die Entwurfssprachen noch auf eine Vielzahl weiterer ingenieurwissenschaftlicher Anwendungsgebiete übertragen werden können. So verspricht beispielsweise der simultane digitale Entwurf des Abgassystemes (Produkt) und der Produktionsanlagen (Digitale Fabrik) mit Entwurfssprachen in der Zukunft immenses Potenzial in der ganzheitlichen Optimierung und Automatisierung des Produktlebenszyklus [9].

Für die Entwicklung von Abgassystem erlaubt der Einsatz des hier dargestellten wissensbasierten Entwurfparadigma, speziell in der Off-Highway-Applikation mit ihren geringen Stückzahlen und individuellen Anforderungen, die schnelle Erzeugung und Validierung von Abgassystementwürfen. Die graphenbasierten Entwurfssprachen können in der zukünftigen Anwendung auf maßgeschneiderte und damit effizientere (Abgas-)Systeme führen und erlauben so eine weitere Reduktion der Verbräuche und Emissionen von Verbrennungsmotoren unter gleichzeitiger Minimierung der Entwicklungskosten.

Literatur

- [1] AKERKAR, R. ; SAJJA, P.: *Knowledge-Based Systems*. Jones & Bartlett Publishers, 2010. – ISBN 9781449612948
- [2] ALBER, R.: *Synthese und Evolution einer technischen Entwurfsgrammatik nach dem Vorbild biologischer Wachstums- und Entwicklungsprinzipien*. Diplomarbeit Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 2001
- [3] ALBER, R. ; RUDOLPH, S.: On a grammar-based design language that supports automated design generation and creativity. In: *Proc. of IFIP WG5.2 Workshop on Knowledge Intensive CAD (KIC-5)*, 2002
- [4] ALBER, R. ; RUDOLPH, S.: "43" - A Generic Approach for Engineering Design Grammars. In: *AAAI Spring Symposium Technical Report SS-03-02*, March 24-26, 2003
- [5] ALT, H. ; BLUM, N. ; MEHLHORN, K. ; PAUL, M.: Computing a maximum cardinality matching in a bipartite graph in time $O(\dots)$. In: *Information Processing Letters* 37 (1991)
- [6] ALT, H.W.: *Lineare Funktionalanalysis: Eine Anwendungsorientierte Einführung*. Springer-Verlag GmbH, 1999 (Springer Lehrbuch). – ISBN 9783540439479
- [7] ANDO, Y. ; KIMURA, H.: An Automatic Piping Algorithm Including Elbows and Bends. In: *International Conference on Computer Applications in Shipbuilding*, 2011, S. 153–158
- [8] ANTONSSON, E. ; CAGAN, J.: *Formal Engineering Design Synthesis*. Cambridge University Press, 2001
- [9] ARNOLD, P. ; RUDOLPH, S.: Bridging the gap between product design and product manufacturing by means of graph-based design languages. In: *TMCE*, 2012
- [10] ASH, G.R.: *Dynamic routing in telecommunications networks*. McGraw-Hill, 1998 (McGraw-Hill telecommunications). – ISBN 9780070064140
- [11] ASMARA, A. ; NIENHUIS, U.: Automatic piping system in ship. In: *COMPIT - 5th Int. Conf. on Computer and IT Application Marine Industries*, 2006
- [12] BAEHR, H.D. ; STEPHAN, K.: *Wärme- und Stoffübertragung*. Springer, 2004 (Springer Lehrbuch). – ISBN 9783540401308
- [13] BAI, C. ; GOSMAN, A.: Development of Methodology for Spray Impingement Simulation. In: *SAE International Congress and Exhibition*, 1995
- [14] BAKAN, S.: Klimawandel im 21. Jahrhundert. In: *MTZ-Konferenz - Heavy-Duty, On- und Off-Highway-Motoren, Nürnberg*, 2012
- [15] BALDWIN, Kristen J. (Hrsg.): *Systems Engineering Guide for Systems of Systems*. Department of Defense, United States of America, August 2008
- [16] BALLE, P. ; GEIGER, B. ; KURETI, S. ; ZIRKWA, I.: *Entwicklung eines vanadiumfreien SCR-Katalysators und Implementierung in das SCRT-System zur gleichzeitigen Entfernung von Stickstoffoxiden und Russ aus Diesel-Abgas (VF-Kat)*. HJS Fahrzeugtechnik, Entwicklungsprojekt gefördert durch DBU, AZ 23251 - 21/0, 2008
- [17] BEICHTER, J.: *Systematische Untersuchungen zum Einsatz von Computeralgebrasystemen in Lösungspfadgeneratoren*. Diplomarbeit Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 2010

- [18] BELLMAN, R.: On a routing problem. In: *Quarterly of Applied Mathematics* 16 (1958), S. 87–90
- [19] BENSUADE-VINCENT, Bernadette: Self-Assembly, Self-Organization - A Philosophical Perspective on Converging Technologies. In: *France/Stanford Meeting*, Dec 2006
- [20] BIRKHOFF, F.: *Selektive katalytische Reduktion von Stickoxiden in Kraftfahrzeugen: Untersuchung der Einspritzung von Harnstoffwasserlösung*, Universität Karlsruhe, Diss., 2007
- [21] BIRKHOFF, F. ; MEINGAST, U. ; WASSERMANN, P. ; DEUTSCHMANN, O.: Analysis of the Injection of Urea-water-solution for automotive SCR DeNOx-Systems: Modeling of Two-phase Flow and Spray/Wall-Interaction. In: *SAE World Congress and Exhibition*, 2006
- [22] BIRKHOFF, F. ; MEINGAST, U. ; WASSERMANN, P. ; DEUTSCHMANN, O.: Modeling and simulation of the injection of urea-water-solution for automotive SCR DeNOx-systems. In: *Applied Catalysis B: Environmental* 70 (2007), Nr. 1–4, S. 119–127
- [23] BLAISDELL, J. ; GILB, A. ; PREETHI, P. ; SWEENEY, J. ; VISWANATHAN, K. ; WAY, P. ; ZAMBON, N.: SCR Performance Optimization Through Advancements in Aftertreatment Packaging. In: *Diesel Engine-Efficiency and Emissions Research (DEER) Conference*, 2008
- [24] BLELLOCH, Guy E.: Prefix Sums and Their Applications / Carnegie Mellon University. 1990. – Technical Report
- [25] BOGDANIC, M.: *Simulation von Autoabgasanlagen*, TU Berlin, Diss., 2007
- [26] BÖHNKE, D. ; REICHWEIN, A. ; RUDOLPH, S.: DESIGN LANGUAGE FOR AIRPLANE GEOMETRIES USING THE UNIFIED MODELING LANGUAGE. In: *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC /CIE, San Diego, USA*, August 30-September 2, 2009
- [27] BOLLIG, M. ; LIEBL, J. ; ZIMMER, R. ; KRAUM, M. ; SEEL, O. ; SIEMUND, S. ; BRÜCK, R. ; DIRINGER, J. ; MAUS, W.: Next Generation Catalysts are Turbulent: Development of Support and Coating. In: *SAE World Congress and Exhibition*, 2004
- [28] BÖLLING, M.: *Lösungspfadbasierte Analysen im Entwurf komplexer Systeme*, Universität Stuttgart, Diss., 2013
- [29] BOSCH, S.: *Lineare Algebra*.. Springer-Verlag GmbH, 2003 (Springer-Lehrbuch). – ISBN 9783540001218
- [30] BRAUN, J. ; LEHNEN, R.: Modelling Urea Distributions in SCR-Exhaust Aftertreatment Systems. A Simple Approach. In: *1st European Automotive CFD Conference*, 2003
- [31] BREDBERG, J.: On the Wall Boundary Condition for Turbulence Models, Internal Report 00/4 / Chalmers University of Technology, Goeteborg, Sweden. 2000. – Forschungsbericht
- [32] BRONSTEIN, I.N. ; SEMENDJAJEW, K.A.: *Taschenbuch der Mathematik*. Harri Deutsch, 2008. – ISBN 9783817120079
- [33] BUCKINGHAM, E.: On physically similar systems; illustrations of the use of dimensional equations. In: *Phys. Rev.* 4 (1914), S. 345–376
- [34] BULLOCK, X. L.: *Engineering Design Theory: Applying the Success of the Modern World to Campaign Creation*. United States Army - School of Advanced Military Studies United

- States Army Command and General Staff College Fort Leavenworth, Kansas, AY 08-09, 2009
- [35] BUNUS, P. ; FRITZSON, P.: Methods for Structural Analysis and Debugging of Modelica Models. In: *2nd International Modelica Conference, Proceedings, DLR, Oberpfaffenhofen, Germany*, 2002, S. 157–165
- [36] BUSS, Samuel R. ; KIM, Jin-Su: Selectively Damped Least Squares for Inverse Kinematics. In: *Journal of Graphics Tools* 10 (2004), S. 37–49
- [37] CAGAN, J. ; MITCHELL, W. J.: Optimally directed shape generation by shape annealing. In: *Environment and Planning B: Planning and Design* 20 (1993)
- [38] CASTELLANI, E. ; BRADING, K.: *On the meaning of symmetry breaking in Symmetries in physics: philosophical reflections*. Cambridge University Press, 2003
- [39] CHAMBERLAIN, Bradford L.: Graph Partitioning Algorithms for Distributing Workloads of Parallel Computations / University of Washington. 1998. – Forschungsbericht
- [40] CHEVALIER, C. ; PELLEGRINI, F.: PT-Scotch: A tool for efficient parallel graph ordering. In: *4th International Workshop on Parallel Matrix Algorithms and Applications (PMAA '06), September 7-9, 2006*
- [41] CONG, J. ; FANG, Jie ; KHOO, Kei-Yong: DUNE-a multilayer gridless routing system. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 20 (2001), Nr. 5, S. 633–647
- [42] CONG, Jason ; MADDEN, Patrick H.: Performance Driven Multi-Layer General Area Routing for PCB/MCM Designs. In: *IN PROC. DESIGN AUTOMATION CONF*, 1998, S. 356–361
- [43] CORMEN, T. H. ; LEISERSON, C. E. ; RIVEST, R. L. ; STEIN, C.: *Introduction to Algorithms*. 3. MIT Press and McGraw-Hill, 2009
- [44] COURANT, R. ; FRIEDRICHS, K.O. ; LEWY, H.: *Über die partiellen Differenzgleichungen der mathematischen Physik*. J. Springer, 1928 (Mathematische Annalen, Sonderabdruck aus Band 100)
- [45] CRUTCHFIELD, James P. ; YOUNG, Karl: Inferring statistical complexity. In: *Phys. Rev. Lett.* 63 (1989), Jul, 105–108. <http://dx.doi.org/10.1103/PhysRevLett.63.105>. – DOI 10.1103/PhysRevLett.63.105
- [46] CURIE, P.: Sur la symetrie dans les phenomenes physiques. Symetrie d'un champs electrique et magnetique. In: *Journal de Physique* 3 (1894), S. 393–417
- [47] DANCKERT, B.: Brennverfahren nach 2010 - Schwerpunkte von modernen Gesamtkonzepten für HDD. In: *Heavy Duty, Schwere Diesel- und Grossmotoren im Spannungsfeld verschärfter Umweltbestimmungen und Kundennutzen HdT-Tagung, München-Ismaning*, 28.-29. November 2006
- [48] DANCKERT, B. ; VOGEL, S. ; PFEIFER, A. ; BRÜCK, R.: Modulares SCR-System zur Begegnung der Off-Highway-Variantenvielfalt. In: *Heavy Duty-, On- und Off-Highway-Motoren MTZ-Konferenz, Kiel*, 2011
- [49] DECHTER, R.: *Constraint Processing*. Elsevier Science, 2003 (The Morgan Kaufmann Series in Artificial Intelligence). – ISBN 9781558608900

- [50] DEMTRÖDER, W.: *Experimentalphysik 1: Mechanik und Wärme*. Springer London, Limited, 2008 (Springer-Lehrbuch)
- [51] DEVADAS, M.: *Selective Catalytic Reduction (SCR) of Nitrogen Oxides with Ammonia over Fe-ZSM5*, ETH Zürich, Diss., 2006
- [52] DIJKSTRA, E. W.: A Note on Two Problems in Connexion with Graphs. In: *Numerische Mathematik* 1 (1959), S. 269–271
- [53] DUMAGE, A. L. ; MENDELSON, N. S.: Coverings of bipartite graphs. In: *Canadian Journal Mathematics* 10 (1958), S. 517–534
- [54] DUNGS, S.: *Wissensbasierte Geometriemodelle zur Strukturanalyse*, Uni Duisburg-Essen, Diss., 2008
- [55] ELSNER, Ulrich ; SURVEY, A.: *Graph Partitioning - A Survey*. 1997
- [56] EMIG, G. ; KLEMM, E.: *Technische Chemie: Einführung in die chemische Reaktionstechnik*. Springer, 1995
- [57] ENTLERBERGER, R. ; ENNEMOSER, A. ; FROBENIUS, M. ; BRÜCK, R. ; HENN, J. ; SCHEEDER, A.: Simulation-based development of SCR sprays, Heavy-Duty, On- and Off-Highway Engines. In: *Concepts for the future, 6th International MTZ Conference Kiel*, 2011
- [58] EUROPÄISCHES PARLAMENT, EUROPÄISCHER RAT: *Richtlinie: 2004/26/EG - Mobile Maschinen und Geräte: Gasförmige Schadstoffe*. 2004
- [59] FALCONER, K.: *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, 2004
- [60] FENVES, S.J ; RIVARD, H ; GOMEZ, N: SEED-Config: a tool for conceptual structural design in a collaborative building design environment. In: *Artificial Intelligence in Engineering* 14 (2000), Nr. 3, S. 233 – 247. – ISSN 0954–1810
- [61] FISCHER, S.: *Simulation of the Urea-Water-Solution Preparation and Ammonia-Homogenization with a Validated CFD-Model for the Optimization of Automotive SCR-Systems*, TU Wien, Diss., 2012
- [62] FITCH, R. ; BUTLER, Z. ; RUS, D.: 3D Rectilinear Motion Planning with Minimum Bend Paths / Dartmouth College. 2001. – Forschungsbericht
- [63] FLOYD, R. W.: Algorithm 97: Shortest Path. In: *Communications of the ACM* 5 (1962), S. 345
- [64] FORZATTI, P.: Present status and perspectives in de-NOx SCR catalysis. In: *Applied Catalysis A* 222 (2001), S. 221–236
- [65] FREY, T. ; BOSSERT, M.: *Signal- und Systemtheorie*. Teubner, 2008 (Lehrbuch : Informationstechnik)
- [66] GAMMA, E.: *Design Patterns: Elements of Reusable Object-oriented Software*. Pearson Education, 2004 (Addison-Wesley Professional Computing Series)
- [67] GERLACH, J. ; ROSENSTIEL, W.: Transformationale Entwurfsraum-Exploration. In: *43rd International Scientific Colloquium Technical University of Ilmenau*, September 21-24, 1998

- [68] GEUZAINÉ, C. ; REMACLE, J. F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. In: *International Journal for Numerical Methods in Engineering* (2009). <http://www.geuz.org/gmsh/>
- [69] GIRARD, J. W. ; MONTREUIL, C. ; KIM, J. ; CAVATAIO, G. ; LAMBERT, C.: Advantages of Vanadium SCR Systems for Diesel NO_x Control in Emerging Markets. In: *International Journal of Fuels and Lubricants* 1 (2009), Nr. 1, S. 488–494
- [70] GOLDENFELD, Nigel ; KADANOFF, Leo P.: Simple Lessons from Complexity. In: *Science* 284 (1999), April, Nr. 5411, 87–89. <http://dx.doi.org/10.1126/science.284.5411.87>
- [71] GROSS, J. ; RUDOLPH, S.: Hierarchie von Entwurfsentscheidungen im modellbasierten Entwurf komplexer Systeme. In: *INCOSE TdSE, Tag des Systems Engineerings*, 2011
- [72] GROSS, J. ; RUDOLPH, S.: Dependency Analysis in Complex System Design using the FireSat example. In: *INCOSE International Symposium, Rom*, 2012
- [73] GROSS, Roderich ; DORIGO, Marco: Self-Assembly at the Macroscopic Scale. In: *Proceedings of the IEEE* 96 (2008), Nr. 9, S. 1490–1508
- [74] GRUENWALD, J.: *Verbesserung der Reduktionsmitteldispersion und -verdunstung in SCR-Anlagen*, TU München, Diss., 2007
- [75] GUIRARDELLO, Reginaldo ; SWANEY, Ross E.: Optimization of process plant layout with pipe routing. In: *Computers and Chemical Engineering* 30 (2005), Nr. 1, 99 - 114. <http://dx.doi.org/10.1016/j.compchemeng.2005.08.009>. – DOI 10.1016/j.compchemeng.2005.08.009. – ISSN 0098–1354
- [76] HAJEK, Bruce: Cooling Schedules For Optimal Annealing. In: *Mathematics Of Operations Research* 13 (1988), S. 311–330
- [77] HALL, P.: On Representatives of Subsets. In: *Journal of the London Mathematical Society* s1-10 (1935), S. 26–30
- [78] HALLEY, Julianne D. ; WINKLER, David A.: Consistent concepts of self-organization and self-assembly. In: *Complex*. 14 (2008), November, Nr. 2, 10–17. <http://dx.doi.org/10.1002/cplx.v14:2>. – DOI 10.1002/cplx.v14:2. – ISSN 1076–2787
- [79] HART, P. E. ; NILSSON, N. J. ; RAPHAEL, B.: Correction to: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: *SIGART Newsletter* 37 (1972), S. 28–29
- [80] HAUCK, P.: *Surface chemistry and kinetics of the hydrolysis of isocyanic acid on TiO₂ anatase*, TU München, Diss., 2007
- [81] HAUG, N. ; SCHARER, B.: Minderung von NO_x-Emissionen aus Kraftwerken: Betriebserfahrungen mit der SCR-Technik. In: *Staub, Reinhaltung der Luft* 51 (1991), Nr. 11, S. 389 – 394
- [82] HAZEWINKEL, M.: *Encyclopaedia of mathematics (set)*. Springer, 1994 (Encyclopaedia of Mathematics: An Updated and Annotated Translation of the Soviet "Mathematical Encyclopaedia"). <http://www.encyclopediaofmath.org>. – ISBN 978–1–55608–010–4
- [83] HELD, W. ; EMMERLING, G. ; DÖRING, A. ; RICHTER, K. ; JACOB, E. ; SCHEEDER, A. ; MÜLLER, R. ; BRÜCK, R.: Katalysatortechnologien für schwere Nutzfahrzeuge im EU VI und US2010 Zeitalter; Die Herausforderung der Stickoxid- und Partikelverminderung für zukünftige Motoren. In: *28. Internationales Wiener Motoren Symposium*, 2007

- [84] HERTKORN, P. ; REICHWEIN, A.: On A Model Driven Approach To Engineering Design. In: *Proc. ICED, the 16th International Conf. on Engineering Design*, 2007
- [85] HIRTH, P. ; BRÜCK, R. ; STOCK, H.: Metallische SCR-Katalysatorsysteme; Kenngrößen für optimale Effektivität. In: *FAD Konferenz Dresden*, 2010
- [86] HOLLAND, Alan ; CALLAGHAN, Barry O. ; SULLIVAN, Barry O.: Supporting Constraint-Aided Conceptual Design from First Principles in Autodesk Inventor. In: ORCHARD, Robert (Hrsg.) ; YANG, Chunsheng (Hrsg.) ; ALI, Moonis (Hrsg.): *Innovations in Applied Artificial Intelligence, 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE, Ottawa, Canada, May 17-20* Bd. 3029, Springer, 2004 (Lecture Notes in Computer Science). – ISBN 3-540-22007-0, S. 905-914
- [87] HOPCROFT, J. E. ; KARP, Richard M.: An $n^5/2$ algorithm for maximum matchings in bipartite graphs. In: *SIAM Journal on Computing* 2 (1958), S. 225-231
- [88] HUBER, T. ; WIRBELEIT, F. ; HARTLIEF, H. ; DEHN, J.: Moderne Tools und Methoden in der Entwicklung und Applikation von Verbrennungsmotoren zur Erfüllung der zukünftigen Abgasgesetzgebung. In: *MTZ-Konferenz - Heavy-Duty, On- und Off-Highway-Motoren, Mannheim*, 2010
- [89] HÜTHWOHL, G. ; SCHAIKA, V. ; BLEICKER, D. ; ECKEL, A.: The influence of Urea Droplet Size on the Efficiency of SCR Catalyst for HD-Diesel Engines. In: *AVL International Commercial Powertrain Conference, Graz*, 2009
- [90] IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH: *The Design Compiler 43V2*. www.iils.de. Version: 2005. – Abgerufen: 14.10.2014
- [91] IKEHIRA, S. ; KIMURA, H.: Multi-objective Genetic Algorithms for Pipe Arrangement Design. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, 2006, S. 1869-1870
- [92] IMO - INTERNATIONAL MARITIME ORGANIZATION: *Richtlinie: IMO MARPOL - Marine Pollution - IMO - Annex VI Prevention of Air Pollution from Ships*. May 2005
- [93] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Brochure: ISO International Standards - Today's state-of-the-art global solutions for CEOs*. 2009
- [94] ISO TC 184 SC 4: *STEP - Standard for the Exchange of Product model data (ISO 10303)*, 1984. <http://www.iso.org>. – The STEP is a vendor neutral CAD exchange file format.
- [95] ITO, D. (Hrsg.): *Robot vision: strategies, algorithms and motion planning*. Nova Science, 2009. – ISBN 9781606920916
- [96] ITO, TERUAKI: A genetic algorithm approach to piping route path planning. In: *Journal of Intelligent Manufacturing* 10 (1999), 103-114. <http://dx.doi.org/10.1023/A:1008924832167>. – DOI 10.1023/A:1008924832167. – ISSN 0956-5515
- [97] JACOB, E.: Perspektiven der mobilen SCR-Technik. In: *15. Aachener Kolloquium Fahrzeug- und Motorentechnik, Aachen*, 2006
- [98] KAEFER, S.: *Trockenharnstoff-SCR-System und Betriebsstrategie für Fahrzeuge mit Dieselmotor*, TU Kaiserslautern, Diss., 2004

- [99] KAHN, A. B.: Topological Sorting of large networks. In: *Communications of the ACM* 5 (1962), S. 558–562
- [100] KANG, S.-S. ; M., Sehyun ; HAN, S.-H.: A design expert system for auto-routing of ship pipes. In: *Journal of Ship Production* 15 (1999), S. 1–9
- [101] KARYPIS, G. ; KUMAR, V.: *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs*. Army High Performance Computing Research Center, 1995 (AHPCRC preprint). <http://books.google.de/books?id=E2cgHQAACAAJ>
- [102] KIM, H. ; MIN, K. ; MYUNG, C. ; PARK, S.: Combined experimental and computational approach to improve catalyst flow uniformity and light-off behaviour. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 216 (2002), S. 413–430
- [103] KIRKPATRICK, S. ; GELATT, C. D. ; VECCHI, M. P.: Optimization by Simulated Annealing. In: *Science* 220, 4598 (13 May 1983), 671–680. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.4175>
- [104] KLEIN, M.: *Einführung in die DIN-Normen*. P. Kiehl, DIN Deutsches Institut für Normung e.V., 2007
- [105] KOEBEL, M. ; ELSENER, M. ; KLEEMANN, M.: Urea-SCR: A promising technique to reduce NOx emissions from automotive diesel engines. In: *Catalysis Today* 59 (2000), S. 335–345
- [106] KOEBEL, M. ; ELSENER, M. ; KRÖCHER, O. ; SCHAR, C. ; ROTH LISBERGER, R. ; JAUSSE, F. ; MANGOLD, M.: NOx reduction in the exhaust of mobile heavy-duty diesel engines by urea-SCR. In: *Topics in Catalysis* 30/31 (2004), S. 43–48
- [107] KOEBEL, M ; ELSENER, M ; MADIA, G: Reaction Pathways in the Selective Catalytic Reduction Process with NO and NO₂ at Low Temperatures. In: *Industrial and Engineering Chemistry Research* 40 (2001), Nr. 1, 52–59. <http://pubs.acs.org/doi/abs/10.1021/ie000551y>
- [108] KOECHER, M. ; KRIEG, A.: *Ebene Geometrie*. Springer, 2008 (Springer-Lehrbuch). – ISBN 9783540493273
- [109] KOH, Cheng-Kok ; MADDEN, Patrick H.: Manhattan or non-Manhattan?: a study of alternative VLSI routing architectures. In: *Proceedings of the 10th Great Lakes symposium on VLSI*, ACM, 2000 (GLSVLSI), S. 47–52
- [110] KORMEIER, T. ; RUDOLPH, S.: On self-similarity as a design paradigm. In: *Proceedings of IDETC/CIE, International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, September 24–28, 2005
- [111] KORMEIER, Thomas: *Graphenbasierte Entwurfssprachen zur konsistenten Modellierung und musterbasierten Topologiemodifikation von Faserverbundstrukturen*, Universität Stuttgart, Diss., 2010
- [112] KOZEN, C.: *The Design and Analysis of Algorithms*. Springer-Verlag Inc., 1991
- [113] KRÖCHER, O.: Harnstoff-SCR für Dieselmotoren: Von der Grundlagenforschung zur Anwendung im Nutzfahrzeug. In: *VöV-Fachtagung "Technik und Betrieb Bus"*, 2005
- [114] KRÖPLIN, B. ; RUDOLPH, S.: Entwurfsgrammatiken - Ein Paradigmenwechsel? In: *Der Prüflingenieur* 26 (2005), S. 34–43

- [115] LAM, Jimmy: *An Efficient Simulated Annealing Schedule*, Department of Computer Sciences, Yale University, Diss., 1988
- [116] LATOMBE, J.C.: *Robot Motion Planning*. Springer, 1990 (The Springer International Series in Engineering and Computer Science). – ISBN 9780792391296
- [117] LAWSON, B.: *How Designers Think*. Taylor & Francis, 2012. – ISBN 9781136398001
- [118] LEE, C. Y.: An Algorithm for Path Connections and Its Applications. In: *IRE Transactions on Electronic Computers EC-10* 2 (1961), S. 346–365
- [119] LORENZ, E.: Predictability: Does the Flap of a Butterfly’s wings in Brazil Set Off a Tornado in Texas? In: *AAAS Section on Environmental Sciences New Approaches to Global Weather: GARP, 139th Meeting*, Dec 1972
- [120] LUGER, George F. ; CHAKRABARTI, Chayan: Knowledge-Based Probabilistic Reasoning from Expert Systems to Graphical Models / University of New Mexico. 2009. – Forschungsbericht
- [121] MATTES, W.: The BMW Approach for Tier 2 Bin 5. In: *Diesel Engine-Efficiency and Emissions Research (DEER) Conference*, 2007
- [122] MEDHI, D. ; RAMASAMY, K.: *Network Routing: Algorithms, Protocols, And Architectures*. Elsevier/Morgan Kaufmann, 2007 (The Morgan Kaufmann Series in Networking). – ISBN 9780120885886
- [123] MEDJDOUB, B.: Constraint-based adaption for complex space configuration in building services. In: *Journal of Information Technology in Construction* (2009), S. 153–158
- [124] MEHLHORN, Kurt ; NÄHER, Stefan ; UHRIG, Christian: LEDA: A Platform for Combinatorial and Geometric Computing. 38 (1999)
- [125] METROPOLIS, Nicholas ; ROSENBLUTH, Arianna W. ; ROSENBLUTH, Marshall N. ; TELLER, Augusta H. ; TELLER, Edward: Equation of State Calculations by Fast Computing Machines. In: *The Journal of Chemical Physics* 21 (1953), Nr. 6, S. 1087–1092. – ISSN 00219606
- [126] MIEDE, A. ; REPP, N. ; ECKERT, J. ; STEINMETZ, R.: Self-Organization Mechanisms for Information Systems - A Survey. In: *Association for Information Systems (AIS): Proceedings of the Fourteenth Americas Conference on Information Systems*, August 14th–17th 2008
- [127] MOLLENHAUER, K. (Hrsg.) ; TSCHÖKE, H. (Hrsg.): *Handbuch Dieselmotoren*. Springer, VDI-Buch, 2007
- [128] MURPHY, C. ; YATES, J.A.: *The International Organization for Standardization (ISO): global governance through voluntary consensus*. Routledge, 2009 (Global institutions series). <http://books.google.de/books?id=iXAeAQAAIAAJ>. – ISBN 9780415774284
- [129] MURRAY, Richard M. ; LI, Zexiang ; SASTRY, S. S.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994
- [130] NEUNDORF, W.: Polynome, Interpolation, Splines und Differentiation / Technische Universität Ilmenau, Institut für Mathematik. 2004. (M 16-04). – Forschungsbericht
- [131] NOLTING, W.: *Grundkurs Theoretische Physik 6: Statistische Physik*. Springer, 2005 (Springer-Lehrbuch). – ISBN 9783540205050

- [132] NOLTING, W.: *Grundkurs Theoretische Physik 2: Analytische Mechanik*. Springer, 2010 (Springer-Lehrbuch). – ISBN 9783642129490
- [133] OBJECT MANAGEMENT GROUP, OMG: *Unified Modeling Language, Infrastructure, Version 2.4.1*. www.omg.org/spec/UML/2.4.1/Infrastructure. Version: 2011-08-05. – Abgerufen: 14.10.2014
- [134] OBJECT MANAGEMENT GROUP, OMG: *Unified Modeling Language*. www.uml.org. Version: 2014. – Abgerufen: 14.10.2014
- [135] OGILVY, C.S.: *Excursions in Geometry*. Dover Publ., 1990 (Dover Classics of Science and Mathematics). – ISBN 9780486265308
- [136] OPEN CASCADE SAS, OCC: *Open CASCADE Technology*. www.opencascade.org. Version: 2014. – Abgerufen: 14.10.2014
- [137] OPENFOAM FOUNDATION: *OpenFOAM - Open Source Field Operation and Manipulation*. www.openfoam.org. Version: 2014. – Abgerufen: 14.10.2014
- [138] PAHL, G. ; BEITZ, W. ; FELDHUSEN, J. ; GROTE, K.H.: *Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. Springer, 2003 (Springer-Lehrbuch, 6. Auflage)
- [139] PARK, Jin-Hyung ; STORCH, Richard L.: Pipe-routing algorithm development: case study of a ship engine room design. In: *Expert Systems with Applications* 23 (2002), Nr. 3, 299 - 309. [http://dx.doi.org/10.1016/S0957-4174\(02\)00049-0](http://dx.doi.org/10.1016/S0957-4174(02)00049-0). – DOI 10.1016/S0957-4174(02)00049-0. – ISSN 0957-4174
- [140] PELZ, G.: *Mechatronic Systems: Modelling and Simulation with HDLs*. John Wiley & Sons, 2003. – ISBN 9780470849798
- [141] PIAZZESI, G.: *The Catalytic Hydrolysis of Isocyanic Acid (HNCO) in the Urea-SCR Process*, ETH Zürich, Diss., 2006
- [142] PIAZZESI, G. ; KRÖCHER, O. ; ELSENER, M. ; WOKAUN, A.: Influence of NO₂ on the hydrolysis of isocyanic acid over TiO₂. In: *Applied Catalysis B: Environmental* 65 (2006), S. 169–174
- [143] PISCHINGER, R. ; KLELL, M. ; SAMS, T.: *Thermodynamik der Verbrennungskraftmaschine*. Springer, 2009
- [144] PRUSINKIEWICZ, P. ; LINDENMAYER, A.: *The Algorithmic Beauty of Plants*. Springer-Verlag, 1996 (The Virtual Laboratory). – ISBN 9780387946764
- [145] REDDY, S. Y. ; FERTIG, K. W. ; SMITH, D. E.: Constraint Management Methodology For Conceptual Design Tradeoff Studies. In: *Design Engineering Technical Conferences and Computers in Engineering Conference, August 18-22, Irvine, California, 1996*
- [146] REICHWEIN, A.: *Application-specific UML Profiles for Multidisciplinary Product Data Integration*, Universität Stuttgart, Diss., 2011
- [147] RENYI, A.: On the dimension and entropy of probability distributions. In: *Acta Mathematica Academiae Scientiarum Hungarica* 10 (1959), 193-215. <http://dx.doi.org/10.1007/BF02063299>. – DOI 10.1007/BF02063299. – ISSN 0001-5954
- [148] RICH, B. R.: *Clarence Leonard (Kelly) Johnson - A Biographical Memoir*. National Academies Press, Washington D.C., 1995 (Biographical Memoir)

- [149] ROBERTS, A. J.: Use the information dimension, not the Hausdorff /
url<http://arxiv.org/abs/nlin.PS/0512014>. Version: Juli 2005. <http://arxiv.org/abs/nlin.PS/0512014>. 2005. – Forschungsbericht
- [150] ROGERS, K.: SCR Inlet Maldistributions - Their Effects and Strategies for Their Control. In: *Conference on Selective Catalytic Reduction and Non-Catalytic Reduction for NO_x Control*, 2002
- [151] ROGERS, K.: Mixing Performance Characterization for Optimization and Development on SCR Applications. In: *Conference on Selective Catalytic Reduction and Non-Catalytic Reduction for NO_x Control*, 2003
- [152] ROH, Myung-II ; LEE, Kyu-Yeul ; CHOI, Woo-Young: Rapid generation of the piping model having the relationship with a hull structure in shipbuilding. In: *Adv. Eng. Softw.* 38 (2007), April, Nr. 4, 215–228. <http://dx.doi.org/10.1016/j.advengsoft.2006.10.002>. – DOI 10.1016/j.advengsoft.2006.10.002. – ISSN 0965–9978
- [153] RUBIN, E. S. ; YEH, S. ; HOUNSHELL, D. A.: Experience curves for power plant emission control technologies. In: *International Journal Energy Technology and Policy* 2 (2004), S. 52–69
- [154] RUDOLPH, S.: *Eine Methodik zur systematischen Bewertung von Konstruktionen*, Universität Stuttgart, Diss., 1995
- [155] RUDOLPH, S.: *Übertragung von Ähnlichkeitsbegriffen*. Habilitationsschrift, Universität Stuttgart, 2002
- [156] RUDOLPH, S.: *Persönliche Kommunikation*. 2009-2013. – Institute for Statics and Dynamics of Aerospace Structures, Faculty of Aerospace Engineering, University of Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
- [157] RUDOLPH, S.: Know-How Reuse in the Conceptual Design Phase of Complex Engineering Products - Or: "Are you still constructing manually or do you already generate automatically?". In: *Conference Proceedings Integrated Design and Manufacture in Mechanical Engineering (IDMME), Grenoble, France.*, May 17-19th, 2006
- [158] RUDOLPH, S.: On The problem of multi-disciplinary system design - and a solution approach using graph based design languages, 1st ACCM Workshop on Mechantronic Design, Linz, November 30, 2012
- [159] RUDOLPH, S. ; BÖLLING, M.: Constraint-based conceptual design and automated sensitivity analysis for airship concept studies. In: *Aerospace Science and Technology* 8 (2004), S. 333–345
- [160] RUDOLPH, Stephan ; NOSER, Hansrudi: On Engineering Design Generation with XML-Based Knowledge-Enhanced Grammars. In: *Proceedings IFIP WG5.2 Workshop on Knowledge Intensive CAD (KIC-4)*, 2000, S. 22–24
- [161] SAJJA, P. S. ; AKERKAR, R.: Knowledge-Based Systems for Development. In: *Advanced Knowledge Based Systems: Model, Applications & Research* 1 (2010), S. 1–11
- [162] SALIM, S. M. ; CHEAH, S. C.: Wall y^+ Strategy for Dealing with Wall-bounded Turbulent Flows. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists - IMECS - Vol. II*, March 18-20, 2009
- [163] SATTLER, K. ; KASPER, W.: *Verfahrenstechnische Anlagen*. Willey-VCH Verlag GmbH, 2000

- [164] SCHAEFER, J. ; RUDOLPH, S.: Satellite Design by Design Grammars. In: *Aerospace Science and Technology* 9 (2005), S. 81–91
- [165] SCHAEFER, J. ; RUDOLPH, S.: Satellite design by design grammars. In: *Aerospace Science and Technology* 9 (2005), Nr. 1, S. 81 – 91
- [166] SCHEEDER, A. ; PFEIFER, A. ; HOLZ, O. ; VOGEL, S.: Applikation eines druckluftlosen SCR-Dosiersystems an Liebherr Stufe IIIB / Tier 4i - Motoren für den weltweiten Einsatz in Mobilkranen. In: *FAD Konferenz Dresden*, 2011
- [167] SCHIFFGENS, H. ; BÜLTE, H.: Technologiekonzept für Mobile Arbeitsmaschinen zur Erfüllung der Emissionsstufe 4 - Plattformkonzept und Applikationen. In: *MTZ-Konferenz - Heavy-Duty, On- und Off-Highway-Motoren, Friedrichshafen*, 2009
- [168] SCHMIDT, Jens: *Eine Entwurfssprache für Geometrie*. Studienarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 2012
- [169] SCHUH, G. ; LENDERS, M. ; ARNOSCHT, J. ; RUDOLF, S.: *Effizienter innovieren mit Produktbaukästen Studienergebnisse und Leitfaden - ein Beitrag zu Lean Innovation*. Werkzeugmaschinenlabor WZL der RWTH Aachen, 2010
- [170] SCHÜTTE, T.: *Ablagerungs- und Alterungsverhalten waessriger Harnstoffloesung bei selektiver katalytischer Reduktion von Stickoxidemissionen*, Universität Lüneburg, Diss., 2010
- [171] SERRANO, D. P.: *Constraint management in conceptual design*, Department of Mechanical Engineering, MIT, Diss., 1987
- [172] SERUGENDO, Giovanna Di M.: Engineering Self-Organising Applications. In: *Proceedings of 1st International Summer School on Pervasive Adaptation*, 7-13 September 2008
- [173] SHALIZI, C.: *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*, University of Wisconsin, Diss., 2001
- [174] SHAPIRO, S.C.: *Encyclopedia of artificial intelligence*. Wiley, 1992 (Encyclopedia of Artificial Intelligence Bd. 1). – ISBN 9780471503057
- [175] SIGNER, M.: Euro VI und dann? In: *Landesverkehrskonferenz Mecklenburg-Vorpommern*, 2011
- [176] SIRIGNANO, W. A.: *Fluid Dynamics and Transport of Droplets and Sprays*. Cambridge University Press, 1999
- [177] SMITH, Russell: *ODE - Open Dynamics Engine*. www.ode.org. Version: 2007. – The Open Dynamics Engine (ODE) is a physics engine in C/C++. Its two main components are a rigid body dynamics and a collision detection.
- [178] SORNETTE, D.: *Critical Phenomena in Natural Sciences: Chaos, Fractals, Selforganization, and Disorder : Concepts and Tools*. Springer-Verlag GmbH, 2004 (Springer Series in Synergetics Series). – ISBN 9783540407546
- [179] SOUKUP, J.: Global router. In: *Proceedings of the 16th Design Automation Conference*. Piscataway, NJ, USA : IEEE Press, 1979 (DAC '79), 481–484
- [180] SRIRAM, D. ; STEPHANOPOULOS, G. ; LOGCHER, R. ; GOSSARD, D. ; GROLEAU, N. ; SERRANO, D. ; NAVINCHANDRA, D.: Knowledge-Based System Applications in Engineering Design: Research at MIT. In: *AI Magazine* 10 (1989), S. 79–96

- [181] STARKE, Jens ; ELLSAESSER, Carmen ; FUKUDA, Toshio: Self-organized control in cooperative robots using a pattern formation principle. In: *Physics Letters A* 375 (2011), Nr. 21, S. 2094 – 2098. – ISSN 0375–9601
- [182] STEINBACH, S.: *Einfluss der Transportvorgänge auf die Effizienz von Harnstoffkatalysatoren in SCR-Abgasanlagen*, Technische Universität München, Diss., 2007
- [183] STEKOLSCHIK, A.: *Ein Beitrag zum ganzheitlichen Qualitätsmanagement von CAD-Modellen in der Produktentstehung*. Shaker, 2006. – ISBN 9783832266691
- [184] STROTS, V. ; SANTHANAM, S. ; ADELMAN, B. ; GRIFFIN, G. ; DERYBOWSKI, E.: Deposit Formation in Urea-SCR Systems. In: *SAE International Journal of Fuels and Lubricants* 2 (2010), S. 283–289
- [185] SUDEIKAT, J. ; BRAUBACH, L. ; POKAHR, A. ; RENZ, W. ; LAMERSDORF, W.: Systematically Engineering Self-Organizing Systems: The SodekoVS Approach. In: *Electronic Communications of the EASST* 17 (2009)
- [186] SZLAPCZYNSKI, R.: An Algorithm for Path Connections and Its Applications. In: *The Journal of Navigation* 59 (2006), S. 27–42
- [187] SZYKMAN, S. ; CAGAN, J.: Synthesis of Optimal Nonorthogonal Routes. In: *Journal of Mechanical Design* 118 (1996), Nr. 3, 419-424. <http://dx.doi.org/10.1115/1.2826902>. – DOI 10.1115/1.2826902
- [188] TARJAN, R. E.: Depth-first search and linear graph algorithms. In: *SIAM Journal on Computing* 1 (1972)
- [189] THOMPSON, George V.: Intercompany Technical Standardization in the Early American Automobile Industry. In: *The Journal of Economic History* 14 (1954), S. 1–20
- [190] TURAU, V. (Hrsg.): *Algorithmische Graphentheorie*. 2. Oldenbourg Wissenschaftsverlag, 2004
- [191] U.S. NATIONAL BUREAU OF STANDARDS: *IGES - Initial Graphics Exchange Specification (NBSIR 80-1978)*. 1980. – The Initial Graphics Exchange Specification (IGES) is a vendor neutral CAD exchange file format.
- [192] VANCZA, J.: Artificial intelligence support in design: A survey. In: *The International CIRP Design Seminar*, Kluwer, Kluwer Academic Publishers, 1999, S. 57–68
- [193] VDI - VEREIN DEUTSCHER INGENIEURE: *Richtlinie: VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. VDI- Verlag, Düsseldorf, 1993
- [194] VELDEN, C. V. ; BILL, C. ; YU, X. ; SMITH, A.: An intelligent system for automatic layout routing in aerospace design. In: *Innovations System Software Engineering* 3 (2007), S. 117–128
- [195] VERHAGEN, Wim J. ; BERMELL-GARCIA, Pablo ; DIJK, Reinier E. ; CURRAN, Richard: A critical review of Knowledge-Based Engineering: An identification of research challenges. In: *Advanced Engineering Informatics* 26 (2012), Nr. 1, S. 5 – 15
- [196] VOGEL, S. ; DANCKERT, B. ; RUDOLPH, S.: Entwicklung von Abgasnachbehandlungssystemen auf Basis einer graphenbasierten Entwurfssprache. In: *MTZ-Konferenz - Heavy-Duty, On- und Off-Highway-Motoren*, Mannheim, 2010

- [197] VOGEL, S. ; DANCKERT, B. ; RUDOLPH, S.: Wissensbasierte Entwicklung von SCR-Systemen mit graphenbasierten Entwurfssprachen. In: *MTZ Motortechnische-Zeitschrift - Peer Reviewed* (September 2012)
- [198] VOLOSHIN, V. I. (Hrsg.): *Introduction to Graph Theory*. Published by Nova Science Publishers, Inc., New York, 2009
- [199] WAGNER, W.: *Strömung und Druckverlust*. Vogel Buchverlag, 1997
- [200] WALZ, C.: *NO_x-Minderung nach dem SCR-Verfahren: Untersuchungen zum Einfluss des NO₂-Anteils*, Universität Karlsruhe, Diss., 2000
- [201] WARD, A. ; SEERING, W.: Quantitative Inference in Mechanical Design Compiler. In: *Proceedings 1st DTM Conference, Montreal, Canada*, 1989
- [202] WEISWEILER, W. ; STIEGER, D.: *Verfahrenstechnische Optimierung eines Ammoniak-Generators zur Bereitstellung von NH₃ für die katalysierte Stickoxid-Minderung in Diesel-Motorabgasen*. ICT, Universität Karlsruhe, Forschungsbericht FZKA-BWPLUS, 1999
- [203] WHITESIDES, George M. ; GRZYBOWSKI, Bartosz: Self-Assembly at All Scales. In: *Science* 295 (2002), S. 2418–2421
- [204] WILBURN, R. ; WRIGHT, T.: SCR Ammonia Slip Distribution in Coal Plant Effluents and Dependence upon SO₃. In: *PowerPlantChemistry* (2004), S. 295–304
- [205] WITKIN, A. ; BARAFF, D.: Physically Based Modeling: Principles and Practice. In: *SIGGRAPH Conference Course, 3.-8. August, 1997*
- [206] WOLF, Tom D. ; HOLVOET, Tom: Emergence Versus Self-Organisation: Different Concepts But Promising When Combined, Springer-Verlag, 2005, S. 1–15
- [207] WOODBURY, R.: Whither Design Space? In: *GCAD '04 International Symposium on Generative CAD System*, July 12-14, 2004
- [208] YU, X. ; GEN, M.: *Introduction to Evolutionary Algorithms*. Springer, 2010 (Decision Engineering). – ISBN 9781849961295
- [209] YUSAN, H. ; RUDOLPH, S.: A study in constraint management integration into the conceptual design phase. In: *Proceedings of DETC'99, ASME Design Engineering Technical Conference*, 1999
- [210] ZÄSCHKE, Tilmann: *ode4j - A Java 3D Physics Engine and Library*. www.ode4j.org. Version: 2014. – Abgerufen: 14.10.2014
- [211] ZHENG, G. ; PALMER, G. ; SALANTA, G. ; KOTRBA, A.: Mixer Development for Urea SCR Applications. In: *SAE Commercial Vehicle Engineering Congress and Exhibition*, 2009

Lebenslauf

Angaben zur Person

Name	Samuel Vogel
Geburtsdatum	24.05.1982
Geburtsort	Bad Waldsee
Nationalität	deutsch

Schulbildung

08.1988 - 07.1993	Eugen-Bolz-Schule GHS Bad Waldsee
09.1993 - 07.2002	Döchtbühl Gymnasium Bad Waldsee
Abschluss:	Allgemeine Hochschulreife

Studium

10.2002 - 09.2007	Studium der Physik, Universität Stuttgart
Abschluss:	Diplom-Physiker

Beruf

10.2007 - 01.2008	Auslandsaufenthalt: Südostasien/Australien
04.2008 - 06.2010	Berechnungsingenieur/Teamleiter, Analytik und Simulation, makon engineering GmbH (jetzt DIF), Tettnang
07.2010 - 03.2013	Leiter Vorentwicklung, Methoden und Analytik, DIF Die Ideenfabrik GmbH, Tettnang
seit 04.2013	Berechnungsingenieur, Forschung und Technologieentwicklung Motoren, MTU Friedrichshafen GmbH (Rolls Royce Power Systems AG), Friedrichshafen
seit 02.2016	Postdoc, Arbeitsgruppe Ähnlichkeitsmechanik, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart

ISBN 978-3-942807-03-6