DISSERTATIONEN

JENS WIMMLER

On the Design of Vehicle Architectures for Driving Dynamics subject to Uncertainty and Feasibility Restrictions

Schriften aus dem Lehrstuhl für Mechatronik

On the Design of Vehicle Architectures for Driving Dynamics subject to Uncertainty and Feasibility Restrictions

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik der Universität Duisburg-Essen
zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften
Dr.-Ing.

genehmigte Dissertation

von

Jens Wimmler
aus
Gera

Gutachter: Univ.-Prof. Dr.-Ing. Dr.h.c. Dieter Schramm
Univ.-Prof. Ing. Massimo Guiggiani

Tag der mündlichen Prüfung: 26.04.2019

DOI: 10.17185/duepublico/70257
URN: urn:nbn:de:hbz:464-20190924-131503-9

“If you don’t know your goal each path may be wrong.”
Konfuzius

FOREWORD

The basis for this research originally stemmed from my passion for developing vehicles with regard to vehicle dynamics that I developed in the Formula Student as a mechanical engineering student. The content of this thesis was created during my time as a PhD student from 2014 to 2017 while I was working in the department for functional design and analysis of vehicle dynamics at the BMW AG in Munich. The research was supervised by the institute for mechatronics of the University of Duisburg-Essen.

I would like to particularly thank my doctoral supervisor Prof. Dr.-Ing. Dr. h.c. Dieter Schramm for providing me the possibility to obtain my doctorate at his institute. Special thanks for his professional and organizational guidance and the valuable feedback during those years. The uncomplicated and goal-oriented cooperation was always a pleasure to me.

Furthermore, I would like to thank Prof. Dr. Markus Zimmermann for supervising me at BMW and sharing his expertise in this special environment of very interesting projects. The numerous joint discussions we had were an enriching experience for me.

I would also like to thank my group leader Dipl.-Ing. Bernd Binkowski and my temporary department chiefs Dipl.-Ing. Peter Dick, Dipl.-Ing. Alexander Karajlovic, and Dipl.-Ing. Jürgen Brack for being demanding and also encouranging me.

I also thank the students I supervised during their placement and thesis for their valuable support. Special thanks to: Dominik Eberl, M.Sc., David Elsner M.Sc., Martin Kundla, M.Sc.

I would also like to thank all of my colleagues at BMW and the University of Duisburg-Essen for the discussions and the support. Special thanks to: Dr.-Ing. Johan Berote, Dr.-Ing. Markus Eichstetter, Dipl.-Ing. Gerhard Fischer, Frédéric Etienne Kracht, M.Sc., Dipl.-Ing. Martin Münster, Sebastian Reicherts, M.Sc., Christian Schulz, B.Eng., Marc-Eric Vogt, M.Sc., Dr.-Ing. Martin Wahle and Amir Zare, M.Sc.

Last but not least, I would like to thank my family for their 28 years of deep support which paved my way to this final stage of formal education. Special thanks also to my partner Vanessa Schneider for her support, patience and tremendous understanding during the time I was finishing this doctoral thesis alongside my occupation as an engineer.

Munich, November 2018 Jens Wimmler

Zusammenfassung

Die Erschließung neuer Wachstumspotenziale vor dem Hintergrund der Steigerung von Profitabilität und Verkaufszahlen erfordert kürzere Entwicklungszeiten bei gleichzeitig steigender Modellvielfalt. Beide Handlungsfelder dienen dem Ziel neue Kunden für die Marke zu gewinnen, welche sich bisher nicht mit dieser identifizieren konnten. Die fahrdynamische Grundauslegung ist dabei besonders herausfordernd, da sie zahlreichen physikalischen Wirkzusammenhängen mit vielen Einflussparametern unterliegt und dabei vielen Anforderungen genügen soll (Kommunalität, Funktion, Gewicht, Kosten und Bauraum). Weiterhin sind in der frühen Entwicklungsphase viele Randbedingungen unklar, dennoch müssen wichtige Auslegungsentscheidungen getroffen werden.

Das Ziel dieser Arbeit ist die Entwicklung einer neuen Auslegungsmethode, welche die Vorteile von qualitativen und quantitativen Methoden kombiniert und mit der die gesamthafte fahrdynamische Grundauslegung von Fahrzeugen und Baukästen transparenter und effizienter wird, um somit zukünftigen Anforderungen an die Automobilentwicklung gerecht zu werden.

Die qualitativen Verfahren basieren auf der Bündelung vorhandenen Wissens über physikalisch technische Zusammenhänge in sogenannten Wirknetzen, in denen systematisch Einstellparameter und Gesamtfahrzeugeigenschaften über mehrere hierarchisch angeordnete Ebenen miteinander verknüpft werden. Die quantitativen Verfahren basieren auf stochastischer Lösungsraumanalyse, Target Cascading und Robust Optimization. Um die Ableitung der Anforderungen von oben nach unten, entlang des V-Models in der Fahrdynamik realisieren zu können, sollte die zu entwickelnde Methode folgende Themen berücksichtigen: Unsicherheit, Realisierbarkeit, Durchgängigkeit, Praxistauglichkeit. Der erste Beitrag dieser Arbeit beschäftigt sich mit der Ableitung von Anforderungen an komplexe Systeme. Im zweiten und dritten Beitrag werden für unterschiedliche Subsysteme zwei verschiedene neue Möglichkeiten präsentiert, die gestellten Anforderungen zu plausibilisieren. Anschließend werden neue Methoden vorgestellt, komplexe hierarchisch zerlegte Systeme entlang des V-Models zu entwickeln, bevor die Methoden für die effiziente Auslegung von Produktfamilien erweitert werden. Abschließend werden die vorgestellten Methoden auf ein komplexes Anwendungsbeispiel der Fahrdynamik angewandt.

Die Basis für die Erstellung der beschriebenen Auslegungsmethode ist ein tief greifendes Systemverständnis über das Fahrzeug und dessen auslegungsrelevante Wirkzusammenhänge. Die Identifizierung dieser Wirkzusammenhänge ist jedoch komplex. Einerseits müssen Anforderungen an das Fahrzeug aus verschiedensten Kategorien erfüllt werden und andererseits besteht eine hohe Anzahl an Eigenschaften, welche dazu genutzt werden können, die gestellten Anforderungen zu erfüllen. In einem ersten Schritt wird das Fahrzeug in verschiedene hierarchische Abstraktionsebenen und Eigenschaftscluster unterteilt. Die Partitionierung der Eigenschaftscluster orientiert sich an der Unterteilung des Systems in verschiedene Subsysteme. Jedes Subsystem ist im Fahrzeugentwicklungsprozess einem unterschiedlichen Expertenteam zugeordnet. Auf eine bestimmte Gesamtfahrzeugeigenschaft nehmen dabei meist mehrere Eigenschaften aus verschiedenen Subsystemen Einfluss, was den Systementwurf erschwert. Nach geschehener Partitionierung des Systems Fahrzeug

wird ein erweiterter Ansatz vorgestellt, die identifizierten Wirkzusammenhänge eines Fahrzeuges mit Hilfe von Machine Learning quantitativ abzubilden. Dazu werden die relevanten Eigenschaften der Subsysteme über skalare Werte parametrisiert und durch Monte Carlo Sampling variiert. Zur Erstellung akkurater Ersatzmodelle werden die Ergebnisse gefiltert. Aufgrund der kurzen Rechenzeit der Ersatzmodelle können Anforderungen effizient in Form von zulässigen Bereichen an alle relevanten Subsysteme abgeleitet werden.

Die tatsächlich realisierten Eigenschaften auf der Subsystem-Ebene können von den jeweiligen Expertenteams durch Anpassung von relevanten Einstellparametern auf der Detail-Ebene beeinflusst werden. Zwischen den Anforderungen an ein einzelnes Subsystem bestehen jedoch zusätzliche Zielkonflikte, weshalb die Subsystemeigenschaften nicht frei und unabhängig voneinander einstellbar sind. Um die Durchgängigkeit des anforderungsbasierten Entwickelns in der Praxis umsetzen zu können, muss bei der Ableitung der Subsystemanforderungen die Realisierbarkeit berücksichtigt werden. Bei komplexen Subsystemen sind für die Überprüfung der Realisierbarkeit eigene Subsystemmodelle notwendig. In dieser Arbeit werden für die Subsysteme Reifen und Fahrwerk verschiedene neue Möglichkeiten vorgestellt, die Realisierbarkeit bereits während der virtuellen Entwicklung effizient zu prüfen. Um die anforderungsbasierte und objektive Reifenentwicklung zu ermöglichen, wird eine neue Methode vorgestellt, Gesamtfahrzeugeigenschaften auf Basis funktionaler Reifeneigenschaften zu berechnen. Nicht realisierbare Bereiche werden dabei mit Hilfe eines Klassifikators von der Lösungsraumsuche ausgeschlossen. Zur Ableitung realisierbarer Anforderungen an das Fahrwerk wird ein schnell rechnendes Ersatzmodell des Subsystems erstellt und in den Prozess der Anforderungsableitung integriert.

Um den Prozess der Anforderungsableitung und Finalisierung eines Designs möglichst effizient zu gestalten, bedarf es intelligenter, aufeinander abgestimmter Algorithmen und Methoden sowie einer möglichst intuitiven Schnittstelle zum Entwicklungsingenieur. Dazu werden ein semi-manueller und ein automatisierter Ansatz vorgestellt. Beide Ansätze haben ihre Vor- und Nachteile. Welcher Ansatz für eine konkrete Problemstellung besser geeignet ist, hängt von vielen Faktoren ab, wie zum Beispiel dessen Komplexität sowie vorhandener Ressourcen und Vorlieben des Entwicklungsingenieurs. Aus den genannten Gründen muss von Fall zu Fall entschieden werden. Auf Basis des automatisierten Algorithmus wird eine Erweiterung zur Auslegung von Fahrzeugplattformen präsentiert.

Die vorgestellten Methoden werden im Anschluss auf komplexe Anwendungsbeispiele aus der Fahrdynamikauslegung angewandt. Realisierbare Anforderungen werden an die zu entwickelnden Subsysteme: Reifen, Fahrwerk, Lenkung und Fahrzeug-DNA abgeleitet. Zur Gewährleistung der Praxistauglichkeit werden zahlreiche Gesamtfahrzeuganforderungen an das stationäre, dynamische, lineare, nicht-lineare und Übergangsverhalten des Fahrzeuges gestellt. Zusätzlich werden die Wechselwirkungen der auszulegenden Subsysteme auf Eigenschaften außerhalb der Fahrdynamik, wie zum Beispiel Lenkleistung, mitberücksichtigt. Nach Anwendung der vorgestellten Methode zur Auslegung eines einzelnen Fahrzeugs, wird diese auch zur gesamthaften Auslegung von Fahrzeugarchitekturen genutzt. Die Fahrzeugarchitekturen basieren auf der gleichen Fahrzeugplattform und teilen einzelne Komponenten miteinander.

Abstract

The development of new growth potentials against the background of increasing profitability and sales figures requires shorter development times while increasing model diversity. Both action fields aim to win new customers for the brand who did not identify themselves with it so far. The layout design of vehicle dynamics is particularly challenging since it is subject to numerous physical cause-effect relationships including a number of design variables and many requirements that must be satisfied (commonality, function, weight, costs and installation space). Furthermore, in the early stage of development, many boundary conditions are still unclear, although important design decisions must be made.

The aim of this work is the development of a new design method that combines the benefits of qualitative and quantitative methods, by which the overall layout design of vehicle dynamics for individual vehicles but also vehicle platforms increases in transparency and efficiency in order to satisfy the future requirements on automotive development.

The qualitative procedures are based on the bundling of available knowledge on physical and technical cause-effect relations within dependency graphs that systematically link design variables and overall objective criteria of the vehicle along certain abstraction levels. The quantitative procedures are based on stochastic solution space analysis, target cascading and robust optimization. In order to derive requirements top-down along the V-model within the field of vehicle dynamics, the method to be developed should consider the following issues: uncertainty, feasibility, consistency, and practicality. First, deriving requirements for complex systems is addressed in this work. In the second and third parts, two different and novel methods to evaluate the feasibility of requirements imposed on the subsystems are presented. Then, new methods to develop complex hierarchically decomposed systems along the V-model are presented, before they are extended for the efficient design of vehicle platforms. Finally, the proposed methods are applied to a complex application example in the field of vehicle dynamics.

The foundation of the described design method is based on a deep understanding of the vehicle as a system and its design-relevant cause-effect relationships. The identification of the cause-effect relationships is complex. On the one hand, requirements from various categories are imposed upon the vehicle, and on the other hand, a large number of design variables may be tuned in order to satisfy the imposed requirements. In the first step, the vehicle is decomposed in different hierarchical abstraction levels and clusters of properties. The partitioning of the clusters is aligned with the separation of the vehicle in different subsystems. Each subsystem is assigned to a different expert team in the vehicle development process. Thereby a particular vehicle objective is affected by multiple properties of different subsystems which complicates the system design procedure. After partitioning the vehicle system, an extended approach for mapping the identified dependencies by machine learning in a quantitative manner is introduced. Therefore, relevant properties of the subsystems are parametrized by scalar values and varied by Monte Carlo sampling. The results are filtered to create accurate surrogate models. Due to the short calculation time of the surrogate models, requirements may be derived as permissible ranges to all relevant subsystem properties very effectively.

The final properties on the subsystem level may be tuned by the related expert teams by adapting design variables on the detail level. Separate conflicting goals occur between the requirements on a particular subsystem, which is why the subsystem properties may not be adjusted freely and independently of each other. In order to ensure consistency of the requirement based development in engineering practice, the feasibility must be considered while deriving requirements to subsystem properties. For complex subsystems, separate subsystem models are necessary to evaluate their feasibility. In this work, different novel approaches for evaluating the feasibility efficiently in the virtual development phase are presented for the tire and suspension subsystem. In order to enable requirement based and objective development of tires, a new method is presented which calculates the overall vehicle properties based on functional tire characteristics. In addition, non-feasible tires are excluded from the solution space computation by applying an additional classifier. A fast calculating surrogate model is created for the suspension system and then implemented into the overall design method to derive feasible requirements on the suspension system.

In order to make the process of deriving requirements and finalizing a design as efficient as possible, smart, synchronized algorithms and methods are necessary as well as a highly intuitive interface to the development engineer. Therefore, both a semi-manual and an automated approach are presented. Both approaches have their own advantages and disadvantages. The more appropriate approach for a particular problem statement depends on many factors, such as the accompanying complexity as well as the existing resources and preferences of the development engineer. Therefore, this must be decided case by case. Based on the automated algorithm, an extended version for developing vehicle platforms is presented.

The presented methods are applied to complex application examples within the field of vehicle dynamics. Feasible requirements are forwarded to the relevant subsystems: tire, suspension, steering, and vehicle-DNA. In order to ensure practicability, numerous overall vehicle requirements are imposed on the steady-state, dynamic, linear, non-linear and transient behavior of the vehicle. In addition, interdependencies of the subsystems to be designed to fields outside of classical vehicle dynamics, such as steering power design, are considered as well. After applying the introduced method for the design of one particular vehicle, it is also applied to the design of vehicle architectures. The vehicle architectures are based on the same vehicle platform and share components with each other.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

notation system:
${}^{l}I_{i}$: solution intervall I of the i'th design variable on the l-level

Symbols

Indices

symbol	unit	description
bad		design misses the requirements
cl		construction level
crit		threshold value to a bad design
d		denotes a particular direction or the number of dimensions on the y-level
ds		design space
good		design satisfies the requirements
i		counting variable on the z-level
j		counting variable on the y-level
k		counting variable on the k-level
lb		lower bound
m		number of design variables on the x-level
max		maximum
min		minimum
n		number of systems
norm		normalized
o		number of objectives on the z-level
rb		robustness box
sb		solution box
si		denotes the i'th system
ss		solution space
t		number of targets on the y-level

symbol	unit	description
ub		upper bound
v		in reference to the absolute/vehicle coordinate system
wt		wheel travel
x		x-direction of a coordinate system
y		y-direction of a coordinate system
z		z-direction of a coordinate system
FA		front axle
RA		rear axle

Greek lowercase letters

symbol	unit	description
$\alpha_{\mu,\mathrm{y,\,max}}$	°	slip angle at maximum lateral grip coefficient of the tire
$\alpha_{\mathrm{Mz,\,max}}$	°	slip angle at maximum self aligning torque of the tire
β	°	body slip angle
$\Delta\beta$	°	difference between the body slip angle and the Ackermann angle
$\lvert\beta(j\omega)\rvert$	–	body slip angle amplification
$arg\,\beta(j\omega)$	°	body slip angle phase shift
$\dot{\beta}/_{\delta_{\mathrm{norm}}}$	1/s	amplification of body slip angle normalized w.r.t. its static value
γ	min	camber
$\Delta\delta$	°	difference between the wheel steering angle and the Ackermann angle
$\delta_{\mathrm{f,inner}}$	°	wheel steer angle of the inner front wheel
$\delta_{\mathrm{f,outer}}$	°	wheel steer angle of the outer front wheel
δ_{h}	°	steering wheel angle
$\Delta\delta_{\mathrm{h}}$	°	difference between the steering wheel angle and the Ackermann angle
$\Delta\delta_{\mathrm{h,95\%}}$	°	difference between the steering wheel angle at 95% of the maximum lateral acceleration and a defined lower lateral acceleration
$\delta_{\mathrm{h,f,swd}}$	–	maximal applicable steering angle factor during swd-maneuver
δ_{t}	min	toe
$\delta_{\mathrm{AM,err}}$	–	Ackermann angle error
μ	–	box size measure normalized w.r.t. the design space
$\mu_{15°}$	–	lateral grip coefficient of the tire at 15 degree slip angle
μ_{rb}	–	size measure of a robustness box normalized w.r.t. the design space

symbol	unit	description
μ_{sb}	–	size measure of a solution box normalized w.r.t. the design space
$\mu_{\mathrm{y,\,max}}$	–	maximum lateral grip coefficient of the tire
$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$	1/N	ratio between degression of lateral grip coefficient and vertical tire load
ρ	–	Pearson's linear correlation coefficient
σ_{y0}	m	relaxation length
φ	–	condensed subsystem performance of a suspension system
φ_{feas}	–	feasibility measure of the solution box
φ_{pos}	–	performance measure of the solution box position
φ_{rb}	–	condensed performance measure of a robust design
φ_{sb}	–	condensed performance measure of a solution box
$\angle\varphi^{\circ}$	$^{\circ}$	phase difference angle between yaw velocity and lateral acceleration at 1Hz
$g\phi$	$\frac{^{\circ}}{\mathrm{m/s^2}}$	gradient roll angle over lateral acceleration
$\lvert\dot{\psi}(j\omega)\rvert$	–	yaw velocity amplification
$arg\,\dot{\psi}(j\omega)$	$^{\circ}$	yaw velocity phase shift
$\dot{\psi}/_{\delta_{\mathrm{h}}}$	1/s	yaw velocity response
$\dot{\psi}/_{\delta_{\mathrm{norm}}}$	–	amplification of yaw velocity response normalized w.r.t. its static value
$\dot{\psi}_{\mathrm{ruts,\,max}}$	$^{\circ}$/s	maximum yaw rate while the passage of wheel ruts

Greek uppercase letters

symbol	unit	description
Δ	variable	change of a quantity
Ω_{ds}	–	design space
Ω_{feas}	–	feasible space
Ω_{fss}	–	feasible solution space
$\Omega_{\mathrm{non\text{-}feas}}$	–	non-feasible space
Ω_{rb}	–	robustness box
Ω^{*}_{rb}	–	allowed region of the robustness box
Ω_{sb}	–	solution box
Ω_{ss}	–	solution space

Latin lowercase letters

symbol	unit	description
$\overline{\boldsymbol{aj}}$	mm	vector containing the distances between kinematic joint $\boldsymbol{a}$ and an arbitrary kinematic joint $\boldsymbol{j}$ of the suspension system
$\lvert a_{\mathrm{y}}(j\omega)\rvert$	–	lateral acceleration amplification
$arg\, a_{\mathrm{y}}(j\omega)$	°	lateral acceleration phase shift
$a_{\mathrm{y,hyst}}$	m/s^2	horizontal hysteresis in the characteristic curve M_{h} over a_{y}
$a_{\mathrm{y,\,max}}$	m/s^2	maximum lateral acceleration
$a_{\mathrm{y,reserve}}$	m/s^2	reserve of lateral acceleration on the rear axle
$\boldsymbol{a} \ldots \boldsymbol{i}$	mm	vectors containing the kinematic joint positions of the suspension system
$\tilde{\boldsymbol{a}} \ldots \tilde{\boldsymbol{i}}$	mm	vectors containing the kinematic joint positions of the suspension system
b	mm	kinematic trail
c_i	–	identifier whether the performance of a subsystem is within the allowed region of the robustness box or not
c_{ARB}	N/mm	anti-roll bar stiffness
d		total number of dimensions on the subsystem level or scrub radius in mm
f	Hz	frequency
f_{scale}	–	scaling factor
h_{CoG}	m	center of gravity height
i_{c}	mm/rev	pinion ratio at centered steering rack position
$\Delta i_{\mathrm{c\text{-}end}}$	mm/rev	maximum difference of the pinion ratio
i_{T}	mm/°	steering linkage transmission ratio
$\boldsymbol{j}$	mm	vector containing the kinematic joint position of an arbitrary kinematic joint of the suspension system
l	m	wheelbase of the vehicle
m_{drill}	Nm	maximum drilling torque
m_{red}	kg	reduced mass of the steering rack
m_{veh}	kg	vehicle mass
p		penalty factor or wheel-load lever in mm
s	–	sensitivity class
t_{FA}	m	track width front axle
t_{RA}	m	track width rear axle
v_{dec}	m/s	power decrease speed of the EPS
w_i	–	weighting factor of the i'th design variable
w_{pos}	–	weighting factor for the position of a robustness box

symbol	unit	description
$\boldsymbol{x}$	variable	vector containing all the design variables on the detail level
$\boldsymbol{x}_0$	variable	optimization vector containing all the design variables on the detail level that are considered to be adapted
$\boldsymbol{x}_{\text{ela}}$	N/mm	vector containing all elasticity/bushing related design variables on the detail level
$\boldsymbol{x}_{\text{kin}}$	mm	vector containing all kinematic/position related design variables on the detail level
$x_{\text{rack,high}}$	mm	steering rack stroke from than on pinion ratio is equal $i_{\text{c}} + \Delta i_{\text{c-end}}$
$x_{\text{rack,low}}$	mm	steering rack stroke until pinion ratio is equal i_{c}
$x_{\text{rack, max}}$	mm	maximum steering rack stroke
$\boldsymbol{x}_{\text{MF}}$	variable	vector containing all magic formula related design variables on the detail level
$\boldsymbol{y}$	variable	vector containing the performance values/design variables on the subsystem level
$\boldsymbol{y}_{\text{ela}}$	variable	vector containing the performance values/design variables related to the elastic behavior of the suspension system
$\boldsymbol{y}_{\text{feas}}$	variable	vector containing the performance values of a particular design that is feasible on the detail level
y_i	variable	performance value i'th element of $\boldsymbol{y}$
$y_{k,\text{norm}}$	–	subsystem performance normalized to its corresponding permissible interval
$\boldsymbol{y}_{\text{kin}}$	variable	vector containing the performance values/design variables related to the kinematic behavior of the suspension system
$\boldsymbol{y}_{\text{rob}}$	variable	robust design on the subsystem level
$\boldsymbol{y}_{\text{ste}}$	variable	vector containing the performance values/design variables of the steering system
$\boldsymbol{y}_{\text{sus}}$	variable	vector containing the all performance values/design variables of the suspension system
$\boldsymbol{y}_{\text{tir}}$	variable	vector containing the functional performance values/design variables (FTCs) of the tire
$\boldsymbol{y}_{\text{DNA}}$	variable	vector containing the performance values/design variables of the vehicle-DNA
$\boldsymbol{z}$		vector containing the performance measures on the system level
z_j	variable	performance value j'th element of $\boldsymbol{z}$

symbol	unit	description
Δz_{CoG}	mm	change in center of gravity height over lateral acceleration
$z_{\mathrm{FA,jacking}}$	mm	vehicle body lift at the front axle over lateral acceleration
$z_{\mathrm{RA,jacking}}$	mm	vehicle body lift at the rear axle over lateral acceleration
$z_{\%}^{\mathrm{bad}}$	variable	vector containing the increase of missed overall design goals on the z-level
$z_{\%,\mathrm{crit}}^{\mathrm{bad}}$	variable	threshold value for a not acceptable increase of bad designs on the z-level

Latin uppercase letters

symbol	unit	description
$\boldsymbol{A}$	–	input/output sensitivity-matrix
B	–	number of bad sample designs
$B_{\%}$	–	ratio between bad and total sample designs
$B_{\%,\mathrm{crit}}$	–	critical ratio between bad and total sample designs
C	–	set of common kinematic joints of the suspension system
$\boldsymbol{C}$	–	constraint matrix for common control links
F_{deficit}	N	steering rack force deficit
$F_{\mathrm{frick,rack}}$	N	friction of the steering rack
$F_{\mathrm{rack,max}}$	N	maximum available steering rack force
$F_{\mathrm{rack,s/2}}$	N	steering rack force at half of its stroke
ΔF_{stat}	N	difference between occurring and permissible steering rack force
$F_{\mathrm{z,f\&r,\,min}}$	N	
$F_{\mathrm{z,\,min}}$	N	minimum vertical tire load at maximum lateral acceleration
$F_{\mathrm{z,RA,\%}}$	–	static longitudinal wheel load distribution rear to front
I		set of independent kinematic joints of the suspension system or interval consisting of two threshold values
$\boldsymbol{I}$	variable	multiple intervals consisting of two threshold values
I_{x}	$\mathrm{kg\,m^2}$	roll inertia of a vehicle
I_{y}	$\mathrm{kg\,m^2}$	pitch inertia of a vehicle
I_{z}	$\mathrm{kg\,m^2}$	yaw inertia of a vehicle
K		set of all kinematic joints of the suspension system or self-steering gradient with unit $\frac{^\circ}{\mathrm{m}/s^2}$
K_{α}	N/°	cornering stiffness
K_{z}	N/mm	vertical stiffness
K_{Mz}	Nm/°	self-aligning torque stiffness
M_{h}	Nm	steering wheel torque

symbol	unit	description
$gM_{\mathrm{h,1m/s^2}}$	$\frac{\mathrm{Nm}}{\mathrm{m/s^2}}$	gradient of the steering wheel torque over lateral acceleration in QSSC maneuver at $a_\mathrm{y} = 1\,\mathrm{m/s^2}$
$gM_{\mathrm{h,hyst}}$	$\frac{\mathrm{Nm}}{\mathrm{m/s^2}}$	gradient of the steering wheel torque over lateral acceleration in WEAVE maneuver at $a_\mathrm{y} = 1\,\mathrm{m/s^2}$.
$M_{\mathrm{h,1m/s^2,manual}}$	Nm	manual steering wheel torque at $1\mathrm{m/s^2}$ lateral acceleration
$M_{\mathrm{h,ay-max}}$	Nm	steering wheel torque perceived by the driver at maximum lateral acceleration
$M_{\mathrm{h,dyn}}$	Nm	dynamic steering wheel torque
$\Delta M_{\mathrm{h,loss}}$	Nm	difference between maximum steering wheel torque and steering wheel torque at maximum lateral acceleration
$M_{\mathrm{h,hyst}}$	Nm	vertical hysteresis in the characteristic curve M_h over a_y
$M_{\mathrm{h,res}}$	Nm	steering wheel torque reserve
$M_{\mathrm{z,\,max}}$	Nm	maximum self-aligning torque of a tire
N	–	total number of sample designs
OR	1/min	oscillation ratio
P_{deficit}	W	steering power deficit
ΔP_{dyn}	W	difference between required and available steering power dynamically
$P_{k\%}$	W	steering power at k percent of steering rack stroke ($k = 15, 30, 60, 95$)
P_{res}	W	steering power reserve
ΔP_{stat}	W	difference between required and available steering power stationary
$P_{\mathrm{EPS,\,max}}$	W	maximum power EPS
R	–	correlation coefficient between design variables and objective criteria
R^2	–	coefficient of determination
$R_{\mathrm{track,\,min}}$	m	minimum turning radius at full steering lock
$R_{\mathrm{track}}\,(\delta_\mathrm{h} = 180°)$	m	turning radius at 180° steering wheel angle
RC_h	mm	roll center height
$gRC_{\mathrm{h,wt}}$	mm	roll center height variation over wheel travel
$S_{T,i}$	–	total-effect Sobol Index of the i'th design variable
$T_{a_\mathrm{y}-\dot{\psi}}$	s	time delay between lateral acceleration and yaw velocity response
T_eq	s	time delay between steering input and yaw velocity
$TLLTD$	–	total lateral load transfer distribution

Abbrevations

abbrevation	description
AI	assessment index
ANN	artificial neural network
AMP	amplification
ATC	analytical target cascading
BEV	battery electric vehicle
CV	characteristic value
vehicle-DNA	refers to the basic layout parameters of the vehicle
DoE	design of experiments
DoF	degree of freedom
EPS	electric power steering
E/E	electric/electronic
FTC	functional tire characteristic
KPI	key performance indicator
MF	Magic Formula
MPP	multi parameter plot
OD	overall tire diameter
OEM	Original Equipment Manufacturer
OW	overall tire width
PH	phase
RSM	respone surface model
TIME	Tire Measurement Procedure
$\angle$	angle

CHAPTER 1

INTRODUCTION

The efficient and effective development of complex systems such as *Vehicle Architectures* is associated with many challenges. In this chapter, those challenges are described and available solution approaches are presented. In Section 1.2 the state-of-the-art is given for:

1. Design for Vehicle Dynamics
2. Design Methods
3. Supervised Machine Learning.

Based on the current state-of-the-art, in Section 1.3 fields of action for developing a contemporary design method are identified. A concept for the solution approach is formulated in Section 1.4, which is followed by a presentation of the structure of the work in Section 1.5.

1.1 Motivation and Context

Approximately 134 years ago, the first vehicle was manufactured. Since then, great technical developments have been made. Some of the most important developments were the safety belt, the crush-collapsible zone, the anti-lock braking system, the electronic stability program, the 4-wheel drive, hybrid- and electric vehicles only to name a few. Although vehicles are becoming increasingly complex systems, the cycle time for new models has decreased over the past years through more efficient processes and computer-aided development. Contemporary challenges of the automobile industry are the development and industrialization of electric mobility as well as providing the customers with more useful services to satisfy their needs which result from our modern and digitalized world. Experts expect that in the next five years the automobile industry will change more quickly than in the past 50 years, and that it is on the brink of a revolution (Dudenhöffer 2016). Therefore,

vehicles are becoming even more complex, and the efficiency in vehicle development must be further increased in order to remain competitive.

Enhancing the efficiency in a vehicle development process requires reducing time-consuming tasks such as building prototypes or performing test drives by utilizing more efficient virtual approaches. A common virtual design approach is numerical optimization. Often, optimization does not consider uncertainties during the development process as well as the feasibility of the proposed design in a sufficient manner. Therefore, time-consuming iteration loops become necessary. In addition, the complexity of the design scope must match the actual design phase appropriately. During the early phase, less design variables should be considered in order to compare different concepts effectively. In the later phase, a more complete set should be considered in order to guarantee the target behavior of the vehicle. In engineering practice, several conflicting goals between different teams exist, which results in time-consuming meetings. In order to further increase efficiencies in the product development process, possibilities for the relaxation of conflicting goals are necessary.

The scope of this work is a novel consistent design method that considers the phase-specific complexity of the vehicle, uncertainties as well as feasibility simultaneously and therefore offers great potential for saving both development time and cost. To consider phase-specific complexity, the scope of design characteristics must be variable. Therefore, the system is decomposed into multiple hierarchical abstraction levels. The hierarchical decomposition allows to reduce complexity in the early development phase and increase accuracy in the late development phase. In order to add information from a more detailed abstraction level, this information must be interchangeable between the system and subsystems. In order to further improve efficiencies in the development of complex subsystems, the dependencies to other subsystems should be reduced in order to allow decentralized development of the subsystem. From an organizational point of view, communication between multiple departments may be reduced and therefore, the engineers can focus on the subsystem design. As a result, less iteration loops are necessary for mutual adjustment of the design variable values.

Uncertainties due to changing boundary conditions during the progressing development phase are one of the main reasons for iterations and late product changes. Nevertheless, variations in industrial production can also lead to differences between the desired and actual behavior. Both challenges may be attended by considering them in the early design phase by the aid of simulations. By deriving requirements as permissible intervals on relevant design variables, instead of optimizing a single design point, the overall system performance is satisfied as long as the permissible intervals are not exceeded. Should the discrepancies of a candidate design exceed the permissible intervals, a flexible and systematic approach must be provided in order to come up with a new solution. When deriving permissible intervals, it is important to ensure their *feasibility* from the beginning. As the less complex scope of design characteristics may lead to unrealistic or infeasible requirements, it is important to make inconsistencies transparent as early as possible in order to save time and costs. Another challenge for vehicle manufacturers is the growing product variety which must be developed at the same time. Therefore, the new method

should also support the development of vehicle platforms such that different vehicles may share components, e.g., a wishbone or a tire. As a result, the components do not need to be designed twice and the volume increases which reduces cost. The essential requirements for a contemporary development process could be represented by the following considerations:

1. *How to decompose a complex system such as a vehicle hierarchically into several abstraction levels (e.g., system, subsystem, detail level) in order to reduce complexity and increase transparency of the relevant cause-effect relations.*
2. *How to achieve the extensibility of the considered complexity along the progressing development process.*
3. *How to arrange the decentralized and independent development of complex subsystems more effectively such that less meetings and iterations are necessary.*
4. *How to map the relations between the relevant system properties in a quantitative manner in order to allow a smooth transition between concept phase and series production.*
5. *How to derive requirements top-down along multiple abstraction levels of the V-model, regarding uncertainty and feasibility in a systematic and efficient manner.*
6. *How to make such a method accessible for the design of vehicle platforms.*

1.2 Current State of the Art

For reasons of interdisciplinarity, the current state-of-the-art is divided into three main topics. First, a review of the design of vehicle systems is given with respect to relevant subsystems. Second, existing design methods developed over the past years are presented. Therefore, brief descriptions are given, pros and cons are discussed and conclusions are drawn. In order to save computational effort, surrogate models are generated based on physical models in this work. Therefore, the third part of the current state-of-the-art covers relevant machine learning techniques.

1.2.1 Design for Vehicle Dynamics

Since the scope of this work is the *Design for Vehicle Dynamics*, the relations between overall vehicle properties and relevant subsystems must be considered in particular. Therefore, the current state-of-the-art for the design of the following subsystems: tires, suspension, and steering is given.

Tires

Tire behavior is complex and may be represented with several models. Since the driving direction of the vehicle is mainly adjusted by the driver using the steering wheel, which

directly results in tire forces, accuracy plays an important role. Therefore, depending on the range of application, different tire models are suitable.

For vehicle dynamics investigations, the *Magic Formula* (MF) tire model is normally used (Pacejka 2006; Hirschberg, Rill, and Weinfurter 2007). The MF-tire model is a semi-empirical model that can be parameterized in several ways. The most popular way is the *Tire Measurement Procedure* (TIME) (TNO 1999) which uses several measured data points to fit the model. However, a real tire in hardware is necessary in order to generate the measurement data. In (Niedermeier, Peckelsen, and Gauterin 2013) a parametrization method that optimizes the MF-parameter based on physical properties of the tire is presented. Nevertheless, the determination of the target tire properties is not covered by the method. In (Niedermeier et al. 2013) the interaction of the tire with the vehicle is optimized. Therefore, a two-step optimization between vehicle properties and *functional tire characteristics* (FTCs) as well as between FTCs and MF-parameters is introduced. The question that remains is: Why not directly optimize the MF-parameters based on overall vehicle targets and calculate the FTCs afterwards in order to forward them to the tire manufacturer? In addition, a disadvantage of providing the tire manufacturer with particular target values is that normally not all of them can be accomplished. So far, no approach that specifies target areas for tire properties exists. All mentioned approaches were applied to simple application examples that lack in complexity compared to the real world.

When specifying subsystems in a practical development process it is important also to sufficiently consider relations to vehicle targets other than vehicle dynamics. Since the subsystems considered in this work also determine the steering force demand during the parking maneuver, requirements derived from this maneuver must be considered as well. Therefore, a separate tire model must be applied as the MF-tire model is not accurate enough for the resulting load cases. The underlying reason is that the MF-tire model was built to reproduce the lateral and longitudinal slip behavior of the tire but not its turn slip behavior, which is vital during parking maneuvers (Ma et al. 2016). Several semi-empirical models focusing on the turn slip behavior of the tire may be found in literature (Pacejka 2006; Guo, Lu, and Ren 2001; Guo and Lu 2007; Kuiper and Van Oosten 2007; Bai, Guo, and Lu 2013). The tire model used in this work is presented in Section 2.2.

For vertical dynamics, the selection of an appropriate tire model depends on the frequency range to be investigated. For higher frequencies up to 120Hz, the FTire model is appropriate (Gipser 1999). However, since vertical vehicle dynamics are not directly considered in this work, the FTire model is not presented more in detail.

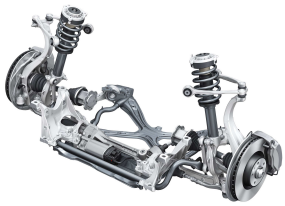
Suspension

The *suspension* system is responsible for maximizing the performance of the tire by influencing its orientation to the road. Depending on the required vehicle behavior it must transfer all tire forces to the vehicle chassis (Gillespie 1992; Blundell and Harty 2004; Schramm, Hiller, and Bardini 2018). Additionally, a certain robustness must be ensured such that the vehicle reacts predictably to different reactions of the driver. Further-

more, the suspension system needs to be designed with respect to geometrical constraints, strength and costs (Heissing, Ersoy, and Gies 2011).

The most common suspension types used in current vehicles are: the Double wishbone, McPherson and Rigid axle in the front and the Rigid axle, Twist beam, Multi-link and Five-link in the rear. Table 1.1 provides an overview of current suspension systems including their advantages and disadvantages.

Table 1.1: Pros and cons of different front suspension systems according to (Heissing, Ersoy, and Gies 2011).

	Double wishbone	**McPherson**	**Multi-link**
			
pros:	braking support angle, kinematic changes during wheel travel, elasto-kinematics, unsprung mass, etc.	unsprung mass, kinematic changes during wheel travel, acoustics, costs	same as double wishbone plus better adjustment of kinematic behavior
cons:	component costs, modularity	Off-Road usability, required steering effort, behavior under acceleration/braking	same as double wishbone
vehicles:	BMW 5-series (G30)	BMW 3-series (F30)	Audi A4 (B9)

The main differences between the individual suspension systems are the number of control links. Basically, it can be said that more control links go hand in hand with more design variables, which offers extended possibilities to resolve conflicting goals. However, it must be considered that the complexity of the design process increases as well.

The requirements on the suspension depend on the overall vehicle targets. In industrial practice, however, the suspension often is not designed requirements-based according to a top-down approach along the V-model but by adapting the concept of a predecessor and evaluating the result, as several reviews of new vehicles confirm (Heissing, Ersoy, and Gies 2011). Radical innovation leaps require different approaches, which was the case when the requirements on the comfort and handling behavior of the vehicle were increased. Back then, the 5-link front suspension was developed. In the future, this radical change could be the extended platform design concept, the shorter development times which would require less manpower or, e.g., the adoption the suspension to the needs of *battery electric vehicles*

(BEVs).

The suspension system itself is a very complex subsystem of the vehicle. Therefore, many design variables must be adapted with respect to many objective criteria during the design process. In addition, each objective criterion is influenced by multiple design variables. As a result, suspension systems are difficult to design and usually require several iteration loops in order to define a final design. Computer-aided design methods related to the field of suspension system design help to reduce development time and costs by handling complexity. Due to the increased available computing power over the past years, numerical optimization is applied more and more often in order to identify the optimal suspension design. To effectively apply optimization techniques, appropriate design objectives and design variables must be defined. In the case of suspension design problems, various parameterizations may be found in literature. Based on the desired input and outcome, an appropriate simulation model must be available. Several design methods may be used in order to find the optimal design in terms of performance, robustness, reliability or a combination of these. In Subsection 1.2.2 an overview of typical design methods is given.

Two main parametrization approaches can be distinguished: *All-At-Once* approaches and *Decomposition* approaches.

In the *All-At-Once* approach, the kinematic joint positions and bushing stiffnesses represent the design variables and are adapted directly and simultaneously in order to fulfill the overall vehicle requirements. Therefore, a full-vehicle model is necessary. Usually, this can be set up with commercial software such as, e.g., Adams/Car, Simpack or IPG CarMaker. When the suspension system is modeled by a multi-body approach, it may be further distinguished between kinematic and elasto-kinematic approaches. However, a disadvantage of this approach is that one model which includes all details is more complex and therefore requires a longer calculation time. In addition, adapting kinematic joint positions with respect to overall vehicle targets such as the yaw velocity response is not intuitive and, in the early design stage, not all detail parameters are already available. Numerical optimization with respect to performance and robustness may help the engineer to find a result that satisfies his requirements. Nevertheless, only requirements that are quantifiable can be considered.

The *Decomposition* approach hierarchically divides the suspension system into two abstraction levels. Therefore, an additional function oriented, map-based level is introduced between the top/vehicle level and the bottom/detail level. The additional intermediate abstraction level complies characteristic maps that summarize the relevant properties of the suspension, e.g., the camber value over wheel steering angle and wheel travel. On the bottom abstraction level of the suspension, the kinematic joint positions and bushing stiffnesses are comprised as described for the All-At-Once approach. As a result, the design problem is divided into two sub-design problems of lower complexity. Therefore, a vehicle model calculates the overall vehicle behavior based on functional suspension characteristics, while a suspension model calculates the functional suspension characteristics based on kinematic joint positions and bushing stiffnesses (Wu et al. 2009). Different map-based vehicle models are presented in (Kvasnicka et al. 2006; Mäder 2012). It can be distinguished between map-based approaches that are measurement-based and those which are

zero-based (Kvasnicka and Schmidt 2010). While measurement-based approaches require existing hardware in order to measure the maps, the zero-based approaches do not. Therefore, in a zero-based approach, the characteristic maps are generated based on scalar values of suspension properties that may be prescribed by the engineer. Typical functional suspension properties are: the camber variation between the minimal and maximal steering angle or the gradient of toe change over wheel travel. For the calculation of suspension characteristics, various suspension models of differing complexity and computation time were introduced during the past years, e.g., (Kracht et al. 2015). Main differences arise from the parametrization depth, the numerical solution algorithm and whether pure kinematic behavior is considered or elasto-kinematics as well. The kinematic joint positions, which serve the input of the suspension model, are usually parameterized as follows: absolute coordinates and/or relative displacements of the kinematic joints (Wimmler et al. 2016), rod lengths plus angles (Rocca and Timpone 2016; Sancibrian et al. 2010) or a combination of both. Bushing stiffness and damping behavior may be mapped by linear (Kracht et al. 2015) or more complex non-linear models (Wolf-Monheim 2014) depending on the predictive statements that should be generated.

The performance of both models may be optimized independently of each other, see Subsection 1.2.2 for an overview of design methods. In the following however, particular examples for the design of suspension systems are presented. Based on sensitivity analysis, the design engineer can get an overview of key design variables in order to adapt particular characteristic curves of the suspension (Rocca and Timpone 2016). Unfortunately, often multiple design objectives are affected by the identified key design variable such that multiple design variables need to be adapted in order to maintain an acceptable subsystem performance. Therefore, multi-objective optimization may help to find an optimal solution in a shorter time period. Sancibrian et al. (2010) present the optimization of a double wishbone suspension using a dimensional synthesis procedure in order to satisfy requirements on the kinematic suspension behavior. In Kang, Heo, and Kim (2010) a robust design optimization is applied to a McPherson suspension system. Therefore, input variables are divided into design variables and random variables. The goal of the optimization is to adapt the design variables such that all design objectives are satisfied, whereas the deviations occurring from the random variables are minimized. The method is called metamodel-based sequential approximate optimization and uses a radial basis function for the approximation of the optimal result. In a sequential process, the metamodel is refined with additional sample points. Although the optimization problem considering three objective criteria was solved, the complexity was not high. Therefore, it is not clear if the presented approach would also converge in the case of more design objectives and how long it would take. Additionally, if the requirements can not be satisfied as a whole, the reusability of the results is not ensured. The identification of the requirements on the suspension system is not treated. Since reducing camber and toe variation over wheel travel does not go hand in hand with a better suspension performance, the practicality of this method is limited.

Although the named approaches serve their purpose and support the design process, the holistic vehicle is not considered sufficiently. In industrial practice, however, this sepa-

rate treatment of particular optimization problems leads to infeasible requirements, which results in several iteration loops. Analytical target cascading, which is described in Subsection 1.2.2, is a hierarchical optimization technique that combines several subproblems.

Steering

The steering system mainly determines the steering feel of the vehicle in combination with the suspension system and the tires. Defining objective overall vehicle targets on the steering feel is difficult for two reasons: First, good steering feel is subjective and defined differently depending on the driver and second, the driver is part of the control loop, and his perception depends on his input (Krüger and Neukum 2001). Nevertheless, previous studies have defined and confirmed common objective criteria to evaluate the steering feel of a vehicle (Harrer 2007; Decker 2009). The results can be summarized as follows: the character and dynamics of a vehicle are strongly perceived subjectively by the driver through the steering wheel response while driving (Harrer 2007). Therefore, in general, three categories are distinguished: steering precision, steering comfort, and steering feedback (Harrer 2007; Braess 2001). Hence, the road-holding properties are perceived through the steering feel as well (Wolf 2009). In this work, the identified objective criteria of the mentioned papers are used in order to derive requirements on the relevant subsystems. The particular objective criteria that are used will be introduced in the respective chapters.

Through the *steering* system, the driver hand torque is transformed to a steering rack force that interacts with the suspension system and the tire in order to adjust a steering wheel angle. A typical steering system of a passenger car is divided into the steering wheel, steering column and steering gear. In 1951, the first vehicle with hydraulic steering assistance was produced by Chrysler (Harrer and Pfeffer 2016). The driver had the possibility of being supported by an additional steering force. The associated additional DoF help to increase steering comfort and performance.[1] As with the suspension system, the steering system needs to be designed in order to meet the overall vehicle requirements. Therefore, the steering system must appropriately support the driver through an additional steering force in any situation as well as providing feedback to the driver on the road surface conditions and vehicle state. In addition, the steering system is a safety-critical subsystem and therefore must be designed with respect to certain legal requirements.

Today, *electric power steering* (EPS) systems are commonly used due to their advantages in weight, applicability, maintenance and fuel consumption to support the driver while steering. Common EPS types are presented in Table 1.2 including their advantages and disadvantages. The *EPS column* (EPSc) has lower requirements in terms of temperature resistance and tightness compared to other concepts, since the servo unit is located inside of the vehicle. However, the location has adverse effects on acoustics and crash behavior. Since the additional force must be transmitted through the entire system to the steering

[1] The steering support of the first hydraulic steering systems was defined by its mechanical and hydraulic transmission ratios. Depending on the steering torque applied by the driver an additional steering force was applied to the steering rack (Harrer and Pfeffer 2016).

wheels, they must be stronger, which increases costs. The *EPS pinion* (EPSp) allows slightly higher forces, since the steering column is not in the load path. Nevertheless, the steering force is limited by the strength of the toothing, and due to the position in the engine compartment, higher requirements on tightness and temperature are necessary. By separating the power paths of the driver and the servo unit, the sprocket transmission is selected separately, which allows the *EPS dual pinion* (EPSdp) greater forces compared to the EPSp. In addition, the crashworthiness may be improved due to the more flexible positioning of the servo unit. If greater forces are required, the *EPS axle parallel* (EPSapa) may be used. Therefore, the EPS-force is transmitted through a belt drive. In addition, the servo unit may be rotated around the steering rack axle which yields in a packaging advantage. The *EPS rack concentric* (EPSrc) features a very compact design on the one hand, but on the other, the increase of the maximum power at constant current results in a larger diameter, since no transmission can be adjusted. Packaging disadvantages are accompanied by the missing possibility to rotate the servo unit around the steering rack.

Table 1.2: Evaluation of different EPS concepts according to (Heissing, Ersoy, and Gies 2011; Harrer and Pfeffer 2016; Bosch 2017; Schramm et al. 2017).

property	**EPSc**	**EPSp**	**EPSdp**	**EPSapa**	**EPSrc**
acousticts	−	+	+	+	+
crash	− −	−	+	+	++
temperature	++	−	−	−	−
tightness	++	−	−	−	−
assembly space	+	−	+	++	−
costs	++	+	−	− −	− −
max. force	− −	−	+	++	++
max. power	−	+	+	++	++

Although a large number of scientific articles that evaluate the advantages and disadvantages of certain EPS-systems on a qualitative level exist, there is a gap in terms of deriving quantitative requirements such as maximum force or maximum power based on overall vehicle targets. Zhang et al. (2008) provide an approach to optimize EPS parameters based on objectified vehicle targets. Nevertheless, the documentation of the approach, e.g., performance function and robustness, is insufficient. In addition, the interaction with the suspension and tire design is not treated, although they have a major influence on the steering system design. Therefore, in engineering practice, trade-offs between the subsystems often must be found with respect to feasibility and cost constraints.

Design variables that are essential for a vehicle with good steering feel are presented in Table 1.3, subdivided into the associated subsystems. However, parameters that have an impact on the architecture design of a vehicle, and therefore are relevant for the layout design, are not specified sufficiently.

As an EPS system is a complex mechatronical system, its design may be clustered in three design fields:

- *mechanical design*, e.g., stiffness, mechanical resistance, durability, acoustics, steering power requirements,
- *electric/electronic (E/E) hardware design*, e.g., circuit boards, microcontrollers, motor design,
- *functional software design and functional safety*, e.g., controllers, algorithms and their integrity.

In contrast to the *mechanical design*, the *E/E hardware design*, as well as the *functional software design and functional safety* can be neglected, since they do not impact the overall vehicle targets considered in this work. However, their design is important for other design objectives. Therefore, recent papers discuss different concepts of control algorithms in order to improve the steering feel of an EPS system (Lunkeit 2014; Uselmann 2017) or the objectification of safety critical-steering disturbances (Fritzsche 2016). The properties of the mechanical design that need to be considered are, for example, the pinion ratio as well as the requirements on the EPS motor. Current literature defines those requirements insufficiently under consideration of the requirements on other relevant subsystems, and thus in assessing the overall vehicle performance in the early development stage. Therefore, cost expensive hardware tests are necessary in order to evaluate the performance of the EPS system in collaboration with the other subsystems. In conclusion, it can be said that the functional design of the mechanical properties of the steering system is far from being well integrated into a requirement based development process.

Table 1.3: Design variables in order to adjust the steering feel according to current literature (Harrer 2007).

Steering System	**Suspension System**	**Tires**
steering ratio	self-aligning torque	drill torque of tires
friction and damping	friction and damping	
servo assistance		
steering elasticity and free play		

1.2.2 Design Methods

Several methods for designing complex systems can be found in literature. This subsection provides an overview of the most relevant methods. Available design methods may be distinguished by their flexibility, robustness, system decomposition, time consumption, information exchange, whether it is a qualitative or quantitative method and several others. Figure 1.1 classifies certain design approaches with respect to their development time and complexity. However, because of increasing computational power, older design methods were developed further and increased in their application bandwidth such that they can

handle complex design tasks as well. Complex design approaches became efficiently applicable due to the increase in computational power. In general, it can be said that the question as to which design approach is the best depends on the design problem itself. In the past, useful design methods were also developed by combining certain design approaches such as target cascading and robust design (Kang et al. 2011) or target cascading and product family design (Fellini et al. 2001). By combining different design approaches to a new one, their particular advantages may be exploited according to the underlying design problem.

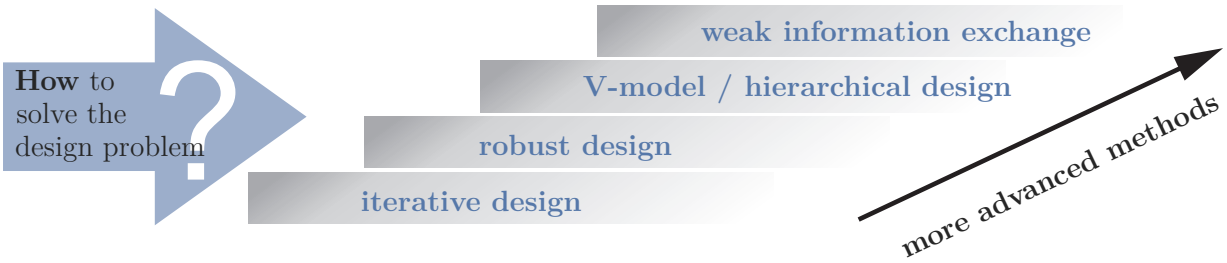


Figure 1.1: Design approaches for solving design problems divided according to their development time and complexity.

Point-Based Design

Point-Based Design or also called Point-Based Engineering designates a design procedure that focuses on one particular design from the beginning. It is distinguished between *Point-Based Serial Design* and *Point-Based Concurrent Design*.

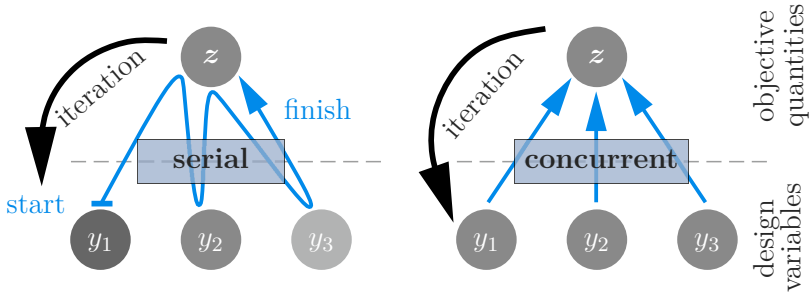


Figure 1.2: Comparison of the Point-Based Serial Design and Point-Based Concurrent Design concept.

Since, *Point-Based Serial Design* is a series of development tasks (e.g., (1) vehicle-DNA[2] → (2) tires → (3) suspension → (4) steering) including very few or no feedback loops, the next downstream task has fewer and fewer possibilities to influence the product to be developed as a whole (Sobek II, Ward, and Liker 1999). Figure 1.2 presents the concept of Point-Based Serial Design. The development procedure starts by specifying the design variable y_1. Thereafter, y_2 and y_3 are adapted based on the target performance measured by the objective criteria $\boldsymbol{z}$. Röski (2012) and Mäder (2012) proposed a Point-Based Serial Design approach in order to reduce complexity and therefore development time in the field of vehicle dynamics.

[2]The vehicle-DNA consists of the basic layout parameters of the vehicle, e.g., vehicle mass or wheelbase, see Subsection 2.2.2.

Contrary to this, in *Point-Based Concurrent Design* development tasks are performed in parallel in order to get early feedback about the system performance before a subsequent iteration step. Therefore, in the case that work tasks are suited to be executed separately, Point-Based Concurrent design enables an agile development process (Johri 2019). A great example from the software-based industry is Amazon. According to Sharon (2018), they release a new production code every 11.6 seconds. However, a problem arises, if tasks cannot be executed separately enough due to interrelations. This may be the case, if changes made by one design team during an iteration loop also affect the results of another design team (Sobek II, Ward, and Liker 1999). In addition, considering cooperation with suppliers, a certain lead time is required depending on the complexity of the product. As a result, Point-Based Concurrent Design can lead to time-consuming iterations and a number of coordination meetings of design teams as well.

Set-Based Design

In *set-based design*, it is possible to distinguish between two main approaches: first, "classical" *set-based design*, and second, *set-based concurrent engineering*.

In "classical" *set-based design*, systems are designed using predefined sets (catalogs) of components and different variations in order to establish feasibility. While the topology of the system is fixed, the particular components may be exchanged with others of the same type. Figure 1.3 contains an example where the product to be developed consists of the components A, B, and C. For each component n-variants are available within the modular system. Therefore, several possibilities exist for composing the product. Within the design space, which is defined by all possible variants of components, regions that are inferior are removed such that the feasible solution space is narrowed. Afterward, sets including design points that are feasible and not Pareto dominated are communicated to the decision makers, while those which are dominated are excluded from the further design procedure. A particular design point is called not Pareto dominated if no design objective may be improved without declining at least one other. Based on the forwarded set of designs, an intersection that meets all qualitative system requirements is sought as well. Infeasibility and Pareto analysis thereafter may help to decide on a particular final design. Ward (1989) first described composing new systems based on predefined sets of components as Mechanical Design "Compiler". Design targets were formulated as permissible intervals (high level "language") and broken down into an optimal design (low level "language"). A design compiler iteratively selects the best suitable components from a predefined set (catalog)

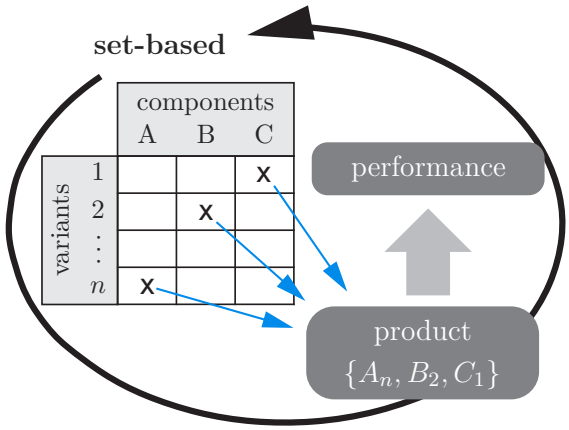


Figure 1.3: Concept of set-based design according to Ward (1989).

by eliminating alternative designs that fail the requirements. In (Finch and Ward 1997) set-based design was combined with predicate logic and constraint satisfaction in order to consider uncertainties as well.

In (Sobek II, Ward, and Liker 1999) *set-based concurrent engineering* (SBCE) is presented as a design method used by Toyota to increase efficiency as well as quality during the development process, while shifting decisions back in the development process. Therefore, engineers communicate sets between each other and multiple design alternatives are developed in parallel. Since flexibility is important for success, especially in the early development phase, a wider set of possible solutions is considered, which is then narrowed until a final decision is made. The framework of SBCE is based on three broad principles: First, map the design space, second, integrate by intersection, and third, establish feasibility before commitment. Although SBCE helps to increase efficiency while working in a big project team, it is more like a structured working method, and therefore the method does not derive quantitative requirements on components.

In (Panchal et al. 2005) an interval-based focalization method for decentralized decision making engineers is presented. The method uses non-cooperative game theoretic protocols in order to establish trade-offs between different design engineers competing for individual goals. The work considers linear and non-linear performance functions by applying the method to two separate design scenarios. When applying the method, first design variables and design objectives are assigned to the design engineers. In the following, an iterative procedure is applied, where each engineer excludes areas of the design space for the design variables that were assigned to them, based on where his design targets cannot be fulfilled. Therefore, the non-cooperative game theory is applied. In addition, a convergence criterion needs to be fulfilled in the first iteration loop in order to find a satisfying solution. Within the remaining intervals, at least one solution that satisfies all requirements is included. Therefore, the method tends to iterate towards a particular solution point, which is an essential difference to the solution space method presented in (Zimmermann and Hoessle 2013), which will be presented in more detail later in this subsection. Since the examples the method was applied to were not very complicated, it remains unknown as to how good the results are for design problems with more than two design variables and design objectives.

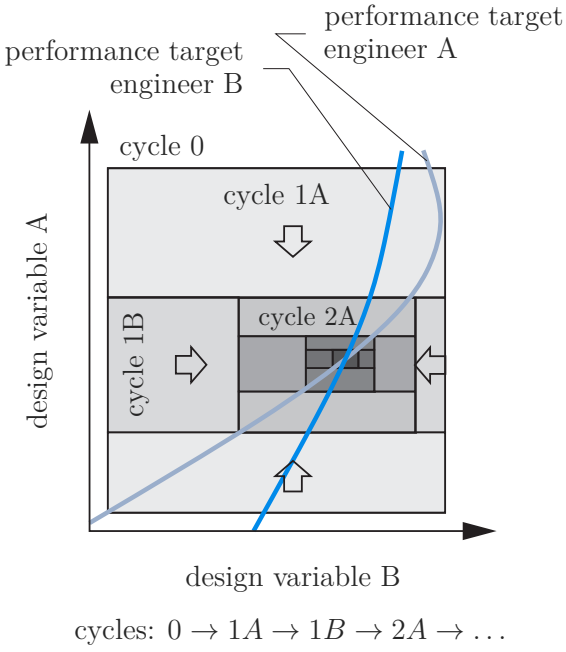


Figure 1.4: Interval-based focalization method for decentralized decision making engineers.

Al-Ashaab et al. (2009) describe a procedure to combine Set-Based Design and the V-

Model approach in order to use the benefits of both. User requirements are captured and then forwarded to several function engineers. After each of them present a set of possible concepts, a new set of concepts that lie within the intersection is selected. The new set is designed in more detail with a special focus on the interfaces of components and system. Before a detailed integration study, a final architecture design is selected. However, this focuses on design principles rather than the quantified derivation of requirements.

Conclusion. Doerry (2012) states that set-based design is good for finding a converged design solution within a defined design space, and that it works well with large design teams. Set-based design helps to reduce expensive back-tracking during a design process by multiple designs during the design process. Although flexibility during the design procedure may be increased, it is vital to stay within the coordinated set of possible final solutions, otherwise this could be a deal breaker. There is a possible trade-off between flexibility and the associated expenses during the design procedure. Since, e.g., three or even more concepts are developed in parallel on a detail level, those personel expenses could also be used for another project. In addition, it can be said that the added value by set-based design is dependent on how good the increased complexity can be handled such that no trade-off between "doing one thing right" and "doing all a little bit" is necessary.

Axiomatic Design

Axiomatic Design is a systematic design methodology and was developed in the late 70s by Nam Pyo Suh (Suh 1979). Requirements and solutions are divided into four domains:

1. customer requirements ($\boldsymbol{CR}$),
2. functional requirements ($\boldsymbol{FR}$),
3. design parameters ($\boldsymbol{DP}$), and
4. process variables ($\boldsymbol{PV}$).

The design process starts at 1. and proceeds in sequence up to domain 4., while each of the following domains presents the solution corresponding to the requirements of the previous one. Therefore, it can help to divide the system into several hierarchies. In order to obtain a better overview of the system architecture, junction-module diagrams and flow diagrams are usually used. The design process is guided by the Independence and Information Axiom. While the Independence Axiom forces the engineer to formulate requirements as well as design parameters such that one design parameter may be adjusted in order to satisfy the requirements without affecting other requirements, the Information Axiom requires that the information is kept short, if possible. The mathematical relation between requirements and design variables is given by $\boldsymbol{FR} = (\boldsymbol{A})\,(\boldsymbol{DP})$, where $\boldsymbol{FR}$ is a vector representing the functional requirements, $\boldsymbol{DP}$ is a vector containing the design variables and $\boldsymbol{A}$ is the design matrix. It is distinguished between uncoupled designs which have a diagonal $\boldsymbol{A}$-matrix, and, decoupled designs which have either an upper or lower triangle $\boldsymbol{A}$-matrix. Using the information axiom, designs that satisfy the independence

axiom can be compared in order to reduce complexity. Probability distributions of the design variables are necessary for calculating the amount of information for a particular design. In (Ge, Lu, and Suh 2002) Ge, Lu and Suh present an axiomatic approach for target cascading. Therefore, axioms guide the decompositions of objectives, integration of a meta-modeling tool and a direct synthesis method.

Conclusion. Although Axiomatic Design provides some useful qualitative approaches for the design of the basic concept of a product many are obvious. Therefore, defining and tracking tasks within a design process, when the necessary knowledge about the system architecture, requirements, functions, relations, etc. is present, or identifying requirements first and then corresponding design variables is not a big challenge. However, there are some useful approaches for the documentation of knowledge. The description of the relations between design variables and objectives via a single Design Matrix $\boldsymbol{A}$ as presented in (Suh 1998) seems to be cumbersome. Especially for systems with many design objectives and design variables where particular design objectives are influenced by multiple design variables, this is not a practical approach. As a "real world" example from the field of vehicle dynamics, the maximum lateral acceleration is mainly influenced by the lateral grip of the tires, however, many other design variables with non-linear interacting sensitivities can be identified (e.g., tire degression, compliance in the suspension system, anti-roll-bar stiffness). Therefore, all design variables must be integrated into the design matrix, since they influence whether the overall requirements are satisfied or not. As a result, the vehicle as a product fails the independence Axiom in many ways which probably is not possible to avoid at all. Nevertheless, selecting design variables in order to reduce dependencies of the design variables also has advantages for other design methods. Another important note is that based on pure Axiomatic Design itself and without combining it with any other method, such as numerical optimization or trial and error, etc., no quantified value or range can be assigned to the design variables.

Robust and Reliable Design

In engineering design, many cases can be found where not only performance requirements are specified to a product, but also requirements on their *robustness* and *reliability*. The reason for that is that should the performance requirements on the product be satisfied in any situation, the customer will be satisfied as well. Each product is subject to uncertainties. It may be distinguished between the following two:

- *Aleatory uncertainties*: also called statistical uncertainties, are uncontrollable. They are associated with parameters and boundary conditions that will always scatter within a certain interval, such as manufacturing-, environmental-, operational conditions and conditions of use. A particular example of aleatory uncertainties are deviating kinematic joint positions of a vehicle due to manufacturing and assembling inaccuracies. (Eichstetter 2017)

- *Epistemic uncertainties*: also called systematic uncertainties, are controllable. These are unknown parameters that could be implemented in the design process, e.g., a

> faster calculating simulation model comes along with less detail and therefore increased epistemic uncertainties. However, they can be reduced by use of a more detailed model. (Eichstetter 2017)

The mentioned uncertainties can lead the product to fail the performance targets even if they were reached in the nominal state. Although robustness and reliability requirements have the goal to increase the quality of the product, they are different. While robustness measures the variation of the performance measure for a given variation of the design parameter values, reliability refers to the failure probability of the system.

Robust Design Several definitions of robust design with almost the same meaning can be found in literature, e.g., (Taguchi, Chowdhury, and Wu 2004; Suh 2001; Box and Fung 1994). The target of robust design is to design a product in a way such that it is insensitive toward aleatory uncertainties. As a result, the required system performance shall also be satisfied, if design parameters and boundary conditions vary within the considered specification. Compared to performance-oriented numerical optimization, a final design point is sought within the design space not only in order to maximize the performance but also to reduce the sensitivity of certain noise parameters against variations. Therefore, the sensitivity of relevant design variables is analyzed and subsequently reduced according to the design problem.

Gerd Gigerenzer (psychologist, Max Planck Institute for Human Development, Berlin) described the need for robustness in relation to the financial world very well. He said that there is no need for decision strategies that are optimal, since optimality is always measured relative to the familiar targets. Instead, robust decision strategies are more important, since they increase the probability that current decisions also will be good from the future perspective. Many fields of science concentrate on optimality, but optimality can only be proven if all assumptions are correct. If only one assumption is wrong, the theoretical optimum will differ from the real world. Therefore, optimality refers to past experiences and does not necessarily need to be true in the future.

According to (Park et al. 2006), robust design can be separated into three different concepts: 1) the Taguchi method, 2) robust optimization, and 3) robust design with the axiomatic approach.

Taguchi Method. The *Taguchi Method* was introduced by Dr. Genichi Taguchi in the late 1940s who said: "Cost is more important than quality but the quality is the best way to reduce cost."[3]. In order to establish robustness, Taguchi applies the following five tools:

1. The P-Diagram divides the variables into a performance target (also known as signal factors) m, noise factors $\boldsymbol{z}$, design variables (control factors) $\boldsymbol{b}$, and performance variables $\boldsymbol{y}$ (Park et al. 2006).

2. An ideal function $\boldsymbol{y}(m)$ describes the quantitative relationships between signal variables m and performance variables $\boldsymbol{y}$.

[3] According to oral tradition.

3. A quality loss function quantifies the loss of customer satisfaction depending on the deviation between performance target m and achieved system performance $y = m \pm \Delta$, where Δ represents the performance deviation due to noise factors. Therefore, usually the quadratic loss function $L = k(y - m)^2$ is applied, where k is the proportionality constant and y is the performance value of the current design, see Figure 1.5.

4. The Signal-to-Noise ratio is used to predict the robustness of a design, the greater the ratio, the more robust the design is. Depending on the design problem type, the Signal-to-Noise ratio is calculated differently. The following three problem types must be distinguished: nominal is the best, the smaller the better, the bigger the better (Park et al. 2006). The example in Figure 1.5 shows the design problem type nominal is the best.

5. Orthogonal arrays are used to identify the optimal setting of the control variables in order to increase the robustness.

The characteristic curve of the probability density φ represents the system performance of a particular design point influenced by noise. In Figure 1.5, two different designs with different probability density functions are given for the same system. Under ideal conditions, without any noise influences, both designs achieve the target performance m. However, the initial design a) offers less robustness. In order to increase the robustness of design a), Taguchi developed a two-step design strategy. In step 1 the Signal-to-Noise ratio is maximized by neglecting the performance objectives. In step 2, the system performance is optimized according to the corresponding design problem type such that the least possible decrease in robustness occurs.

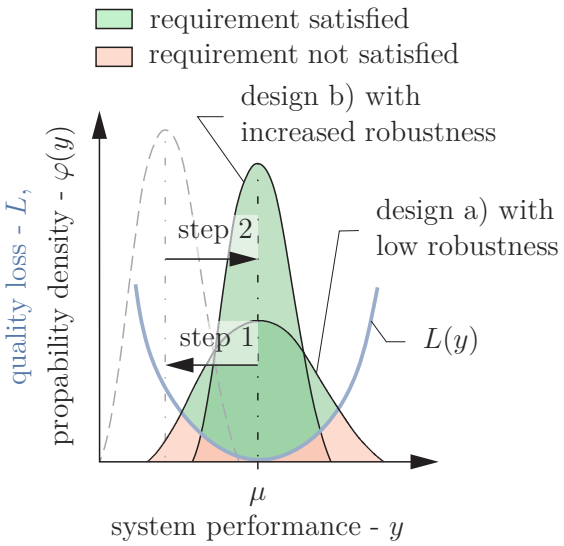


Figure 1.5: Robust Design by Taguchi method.

As a result, design b) is established satisfying the performance requirements at its mean value along with an increased robustness against noises. Therefore, design b) has a decreased probability that the customer will not be satisfied with the product performance at all.

In the original Taguchi Method it is assumed that within step 2 the mean value may be shifted by a single design variable in order to adapt the performance without decreasing robustness. In process design this variable may be time, however, in product design often no such shifting variable exists which in result prevents the direct application of the Taguchi method.

Since the Taguchi Method works with orthogonal arrays, it is a discrete design method. Therefore, a great deal of research has been done to reduce the necessary number of experiments in order to reduce computation time. Further research has been done for applying the method to design problems with multiple characteristic curves. In general, weighting factors were used to combine the results of the characteristic curves (Park et al. 2015; Roy and Chakraborty 2015; Casalino and Pastrone 2015). Although weighting factors are useful in order to prioritize the design goals, they come along with the drawback that the engineer must decide which goal is more important to him at the beginning of the design procedure.

Robust Optimization. Numerical *robust optimization* provides many advantages compared with the original Taguchi Method. Therefore, constraints may be handled easily and design variables that affect the mean performance and the robustness do not need to be separated as they can be considered simultaneously. Using robust optimization, it is possible to improve the robustness of the objective function as well as the robustness of constraints (Cook, Jarrett, and Willcox 2017) such that both are satisfied for all considered noises.

Therefore, the mean standard deviation between the system performance $\mathrm{f}(\boldsymbol{x},\boldsymbol{\xi}_i)$ and the target value $\hat{y}$ is minimized as shown in Equation (1.1), (Beyer and Sendhoff 2007). The overlapping of the input signals with the noise variables $\boldsymbol{\xi}_1 \ldots \boldsymbol{\xi}_\kappa$ leads to increased robustness.

$$\underset{\boldsymbol{x}}{\text{minimize}} \; \frac{1}{\kappa} \sum_{i=1}^{\kappa} (\mathrm{f}(\boldsymbol{x},\boldsymbol{\xi}_i) - \hat{y})^2 \tag{1.1}$$

Several sensitivity measures between input variables and design objectives were implemented in performance functions in order to maximize the robustness (Kitayama and Yamazaki 2014; Gong et al. 2017). Since the computational effort for computing the sensitivity measures can be very expensive, depending on the number of design variables and the sensitivity index itself, *response surface models* RSMs can be used to calculate them (Shahraki and Noorossana 2014). However, the quality of the results also depends on the accuracy of the response surface model.

Robust Design with the Axiomatic Approach. The axiomatic design method may also be applied in order to maintain a robust product. Therefore, in a first step the independence axiom is used in order to define design variables. When several designs that satisfy the independence axiom exist, the information axiom is used for identifying the best one. In terms of the information axiom, the design with the least information exchange is the best. Therefore, the information content may be defined as the probability that the particular design will satisfy the requirements under influence of noise. In general, it can be said that the information content is small, if the mean performance of the product is coincident with the required performance, and the performance variance is also small. Therefore, the information axiom should lead to a similar result as the Taguchi Method.

Six Sigma Approach. In order to enable fault-free production under varying manufacturing tolerances, a certain variability of the design parameters should be provided before

the system performance becomes critical. The *Six Sigma approach* helps to define a necessary tolerance of the design variables. Therefore, based on the probability distribution of a particular design variable, the standard deviation Sigma (σ) is identified (Pande, Neuman, and Cavanagh 2000). Since a product often consists of multiple subsystems with independent tolerances, a spread of the allowed manufacturing tolerance of $\pm\, 6\,\sigma$ has proven successful for fault-free production. Therefore, the product must be designed such that the specification limits have a distance of six standard deviations in each direction. Although the method is statistically proven correct it is difficult to apply in product design, since tolerances need to be tight in order to produce high-quality products, and a low spread of manufacturing tolerances comes along with cost-intensive processes. Therefore, the six sigma approach focuses more on the manufacturing process than on product design.

Reliable Design In reliability-based design, it is assumed that uncertain input variables influence the system performance as in robust design (Stapelberg 2009). In order to avoid failing the system requirements in the manufactured product, uncertainties need to be considered during the design process. However, in reliability based design, a particular design point is sought such that the design goals are satisfied with a prescribed minimum probability. The reliability is quantified by the reliability index which indicates the probability that the design will satisfy the design requirements (Zhao and Ono 1999). Therefore, design for reliability is different to performance optimization, since a deviation through an uncertain parameter leads to the probability that the design will meet the requirements. Instead, if reliability is not considered, deviations may lead to the failure of the product immediately. The measuring range of the reliability index lies between zero and one. In order to calculate the reliability index, all information about uncertain parameters must be known, e.g., whether an input variable is distributed evenly or normally.

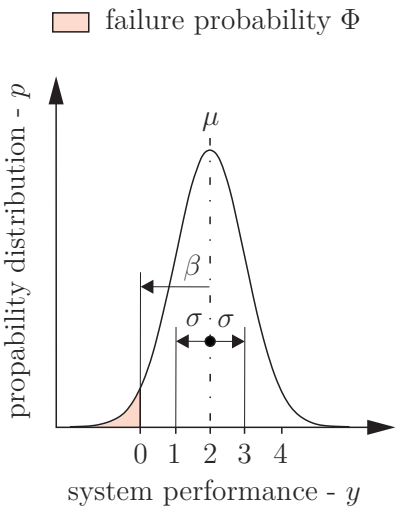


Figure 1.6: Reliability-based design.

As shown in Equation (1.2), the reliability of a design is calculated by one minus the failure probability. The failure probability Φ may be determined by the distribution function and is a function of the safety index β, which is defined by the ratio between the mean value μ and the standard deviation σ, see Equation (1.3).

$$\text{reliability} = 1 - \Phi(-\beta) \tag{1.2}$$

$$\beta = \frac{\mu(g)}{\sigma_g} \tag{1.3}$$

Figure 1.6 presents a simple example of a reliability-based design approach. Designs with a system performance $y > 0$, and therefore meet the design objective, are denoted as good designs, while the others are bad designs that fail the design objective. The mean value μ for the presented example is equal two. Taking the uncertain parameters into account, the occurring system performance is distributed normally. Therefore, the safety index β is equal two for $\sigma = 1$ with the resulting failure probability $\Phi = 0.023$. This means, that about every 45^{th} product produced does not satisfy the requirements.

In the real world, however, system responses often are non-linear, and therefore Φ cannot be calculated based on the distribution function. To overcome this limitation, the first-order-reliability method (FORM) and the second-order-reliability method (SORM) were developed. Both methods calculate the minimum distance between the current design point $\boldsymbol{x}$ and the limit surface $g(x)$ within the normalized design space u-space (also known as Standard Gaussian space) and then estimate Φ. The only difference between FORM and SORM is the representation of the limit surface. While in the FORM, the limit surface $G(\boldsymbol{u})$ is linearized by a 1^{st}-order polynomial, the SORM, therefore, uses a 2^{nd}-order polynomial. Due to the normalization of the input space, the current design point $\boldsymbol{x}^*$ is always coincident with the coordinate origin and the system performance is described by $G(\boldsymbol{u})$ ($g(\boldsymbol{x}) \rightarrow G(\boldsymbol{u})$). If the system performance $G(\boldsymbol{u})$ exceeds 0, the requirements on the system are failed, whereas if the performance is below 0 they are satisfied. Therefore, the limiting surface is defined at $G(\boldsymbol{u}) = 0$. The design point on the limiting surface $G(\boldsymbol{u}) = 0$ with the shortest distance to the current design point $\boldsymbol{x}^*$ is referred to as the most probable point of failure (Lopez and Beck 2012). At the most probable point of failure, the performance function is linearized in the FORM. Based on the linearized performance function, the failure probability and hence the reliability of the system may be computed. Depending on the non-linearity of $G(\boldsymbol{u})$, the error in the calculation of the integral for failure probability increases or decreases. If the number of errors is too high, the SORM method can be used for greater accuracy. However, the SORM method requires increased computational effort. Several optimization techniques for finding the most probable point of failure and improving the reliability of a particular design exist and will be discussed in the following.

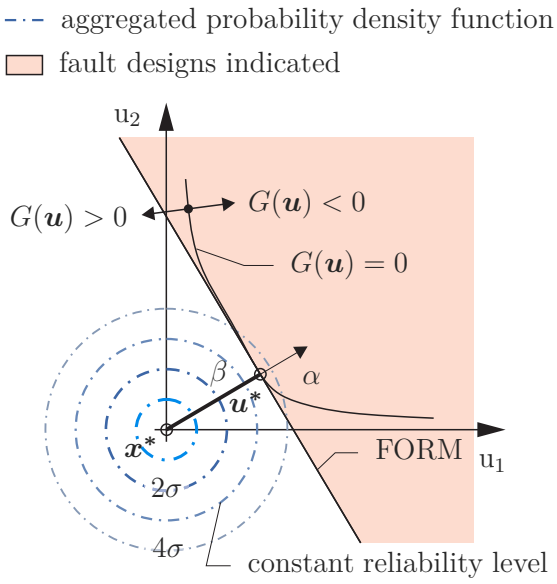


Figure 1.7: Concept of FORM in the normalized input-space.

Reliability-Based Design Optimization. By optimizing a particular design for reliability, the mean values of uncertain variables are adapted under consideration of their

probability distribution with respect to an objective function. The objective function includes probabilistic targets such as the reliability index. Adapting the input variables $\boldsymbol{x}^*$ in order to shift the limit surface in the direction of α increases the safety index β and therefore the robustness of the design. The resulting design not only offers improved performance, but also a higher degree of confidence in achieving the overall design goals. For identifying the optimal design with regard to the objective function, several FORM-based optimization approaches exist. These can be divided into coupled and decoupled approaches. Coupled FORM-based approaches were developed first; they consist of an inner and outer loop which is the reason why they are coupled. The inner loop contains the reliability analysis while the outer contains the design optimization. When coupling two separate iteration loops, the computing expense increases greatly which is the main drawback of this method. However, several decoupling strategies were developed in order to reduce computational expense. Decoupling strategies may be divided into serial single loop methods and unilevel methods and can be further divided into particular methods. Yang and Gu (Yang and Gu 2004) investigated four relevant serial single loop methods and discovered that the single loop single variable (SLSV) method (Chen et al. 1997) is the most effective one. In unilevel methods, the reliability criterion is included in the design optimization (outer loop) as constraints or directly within the performance function. In Cheng et al. (Cheng, Xu, and Jiang 2006) an unilevel strategy was presented which was based on sequential approximate programming (SAP). Yi and Cheng (Yi and Cheng 2008) compared serial single loop (SLSV), unilevel (SAP) and other methods and concluded that in this case, the unilevel strategy leads to the best results. A more extensive overview of different optimization strategies related to reliability based design can be found in (Lopez and Beck 2012).

Conclusion. The reliability-based design considers uncertainties through probability distributions of input parameters. Therefore, all information about the uncertainties must be known. If that is not the case, some probability distributions must be assumed. In addition, the reliability-based design focuses on a particular design point and therefore has some disadvantages compared to the set-based design in terms of flexibility during the ongoing design process. The approximations by the FORM and SORM can lead to inaccuracies that, in most cases, are not crucial.

V-Model Processes

The *V-model* for product development describes a requirement-based design process. Therefore, the system to be designed is divided into several abstraction levels that represent, e.g., system, subsystems, and components. First, the overall system requirements are defined. Then, in a second step, requirements need to be propagated down to the bottom level. In the following, the components are realized and integrated. During the realization and integration, the ultimate properties can be verified with respect to the associated requirements on each level. Since the process steps can be arranged as a "V" over time, the method is called V-model. According to the available literature, the V-model was first mentioned in 1984 by (Boehm 1984) with the intention to improve the software design pro-

cess and emphasize the importance of verification. In 1991 the V-model was first applied to systems design in order to treat the relations between project management and engineering (Forsberg and Mooz 1991). Therefore, the focus was laid on the multilevel and iterative nature of the design process of systems. A uniform Version of the V-model was introduced by Bröhl and Dröschel (1993). Later, it was further developed to the V-model 97 (1997) and V-model XT (2005) (Broy and Rausch 2005), while in 2017 the V-model XT Version 2.1 was introduced (V-Modell-Autoren 2017), see Figure 1.8. In (Haskins 2006) a requirement-based design procedure for systems engineering motivated by the V-model is described in a qualitative manner. In addition, the ISO 26262 (2011) is based on the V-Model and is a reference for designing safety-related E/E systems in series production road vehicles. The V-model has been applied to many design fields such as software, E/E, mechanics design and many more. However, often the process of specifying requirements on the subjacent levels is not defined in a quantitative manner. Therefore, the practical use of the V-model is often limited and several bottom-up iterations become necessary.

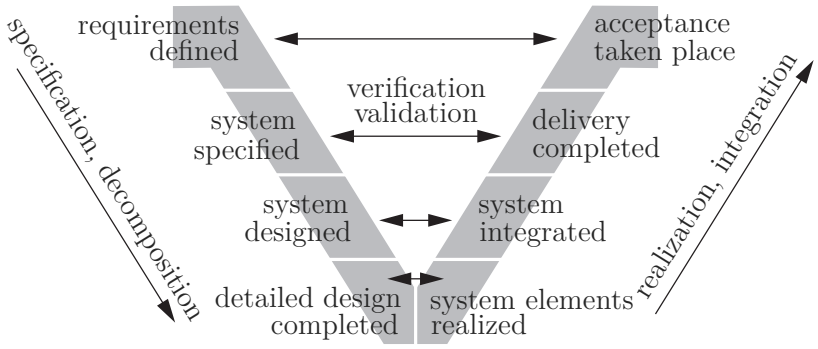


Figure 1.8: System design along the V-Model according to (V-Modell-Autoren 2017).

Conclusion. As mentioned in (Al-Ashaab et al. 2009), the V-model distinguishes between well-defined stages during the development process which is good. However, its sequential nature makes it inflexible to changing requirements and framework conditions. Therefore, in (V-Modell-Autoren 2017) the flexibility was improved significantly. Nevertheless, it focuses on the design procedure and does not provide a quantitative method for deriving and validating requirements on the different abstraction levels. Therefore, in engineering practice, the attempt of requirement-based design often fails or reveals itself as an illusory on closer inspection.

Target Cascading

Typical for *Target Cascading* is that one big design problem is split into several smaller sets of design subproblems on different hierarchical levels. Based on overall system targets, defined as fixed values, a design that minimizes the deviation between target and response is sought on each level. Therefore, a consistent design should be guaranteed while meeting the overall system requirements. The benefits of this procedure are reduced product design-cycle time, avoidance of design iterations late in the development process and increased likelihood that physical prototypes will be closer to production quality (Michelena, Kim, and Papalambros 1999; Michelena et al. 2001; Kim et al. 2003). A

general automated optimization approach for target cascading, including a coordination strategy over any number of abstraction levels, was first introduced by Michelena, Kim, and Papalambros (1999). In 2003 an *Analytical Target Cascading Approach* (ATC) was applied to automotive vehicle design, see (Kim et al. 2003). Later, it was extended for the design of product families, which was the first work that addressed product family design and target cascading in combination (Fellini et al. 2001; Kokkolaras et al. 2002). The extended method was successfully applied to a simple example considering one target for the product family, two vehicles with one additional target each and one subsystem each plus the shared vehicle body. A more complex design problem in the field of chassis design that reviews different design scenarios of one specific vehicle was presented in (Kim et al. 2003). However, the convergence issue in target cascading is further discussed in (Michelena, Park, and Papalambros 2003). In (Kokkolaras, Mourelatos, and Papalambros 2004) target cascading was extended for the consideration of probabilistic design before a probabilistic analytical target cascading formulation was proposed in (Liu et al. 2005). To the development of a suspension concept, ATC was applied by (Wu et al. 2009).

However, the design often is not reliable and modifications in the late development phase are necessary (Ge, Lu, and Suh 2002).

Solution Spaces

The total *solution space* is the region within the design space where all requirements on the product are satisfied. Therefore, those designs are called good. Although multiple papers discuss different solutions within the design space (e.g., (Sobek II, Ward, and Liker 1999)), the quantitative shape of the region in which all possible solutions exist often remains unknown. By calculating solution spaces, multiple issues with classical design methods are solved. The solution space method may be characterized by the following properties:

- *Requirement-based* top-down method that needs target intervals on the top-level as input.
- *Permissible intervals* are derived on the design variables in a quantitative manner.
- *Uncertainties* may be considered without a particular probability distribution.

Since the solution space method is a requirement-based approach with top-level performance targets as input, the required properties of the design variables on the lower levels are defined similarly to performance, robustness or reliability-based optimization. Due to the top-down process, the entire design space may be explored for designs satisfying the overall requirements. Contrary to set-based engineering (Sobek II, Ward, and Liker 1999) or V-Model processes (see Subsection 1.2.2), solution spaces provide a methodology for deriving quantitative requirements with the help of quantitative simulation models. Therefore, solution spaces were applied to practical and application-related design problems which is shown in the following paragraphs. In addition, solution spaces differ from usual numerical optimization, since they consider permissible intervals for design variables instead of particular values. Therefore, solution spaces are fundamentally different. The

underlying idea is to identify and describe the region within the design space that satisfies all requirements under consideration of all constraints. This mind change relaxes conflicting goals while increasing flexibility and robustness. Within the derived permissible intervals, a particular design may be planned in the early design phase, however, when environmental conditions change during the design process it can be switched to an alternative design rapidly. Nevertheless, while selecting a particular design point the following must be considered: the closer the final design to the center of the solution space, the greater the robustness. As a result, no particular probability distribution is necessary in order to increase the robustness of a design. An extensive overview of calculating solution spaces is presented in (Zimmermann et al. 2017).

Computing Solution Spaces Computation of the total Solution Space that includes all designs satisfying the requirements is not practical and also not necessary for two reasons (Zimmermann et al. 2017):

- Due to the arbitrary shape of the total solution space, one can hardly imagine all possible solutions. Especially for high dimensional design problems (four-dimensional and beyond that), the visualization gets more and more difficult.
- Since the system performance often depends on multiple design variables they must be adjusted with respect to each other. However, if the values of the design variables are defined in separate departments they somehow must be decoupled in order to avoid endless iterative adjustment meetings between the departments.

Stochastic solution space algorithm. A stochastic optimization algorithm was introduced by Zimmermann and Hoessle (2013) identifying the box-shaped solution space with the maximum volume. Therefore, based on classical optimization, a first good design that satisfies all requirements is identified. In the following, the design point is replaced by a solution-box that is a superset of the initial design point. Due to an exploration phase, the solution box grows towards the region with the most good designs. To make this possible, three steps are applied iteratively: first, the actual box is extended in each direction, second, Monte-Carlo simulations are performed within candidate box, and third, bad designs are removed. In the following consolidation phase, the confidence level of the candidate solution box is increased by applying the last two steps: removing bad designs and sampling into the candidate box. The optimization is completed when the ratio between bad designs and sample points falls below a certain threshold value. In (Fender et al. 2014), the stochastic algorithm was extended to handle constraints with the intention to reduce the number of components that need to be adapted in order to meet the requirements.

Analytical solution space algorithm. An analytical approach was introduced by Fender (2014). The proposed algorithm is based on tracking box corners and was applied in the context of vehicle crash design. Therefore, classical numerical optimization may be used in order to maximize the volume of the solution box. In (Erschen, Duddeck, and Zimmermann 2015) the analytical approach is extended for general design problems using multiple linearized boundaries of the solution space.

Application of Solution Spaces *Crashworthiness.* In (Song, Fender, and Duddeck 2015) the analytical approach is applied to vehicle crash design considering vehicle architectures that share certain components.

Vehicle Dynamics. In (Eichstetter et al. 2014) the stochastic solution space algorithm is applied to the functional damper design of a vehicle. Permissible intervals are calculated based on overall vehicle targets on the lateral and vertical dynamics. Münster et al. (2014) applied the stochastic solution space algorithm to the vehicle steering design. Considering requirements on the steering wheel torque and steering wheel vibrations, based on desired frequency responses, requirements on physical parameters of the front axle, steering system, and the excitation are derived. For the concurrent design of vehicle tires and axles, Wimmler et al. (2015) derived solution spaces on both subsystems, see Figure 1.9. In addition, the design variables were split into two separate clusters according to the relevant objective criteria. The proposed method is applied to different simple application examples in the field of vehicle dynamics. In addition, the feasibility of requirements on FTCs is ensured by a tire database of already existing tires. In (Eichstetter, Müller, and Zimmermann 2015) solution spaces were extended and applied to product family design in vehicle dynamics. Therefore, overlapping regions of solution spaces indicate where the same component may be shared between different vehicles. The Bell's number is used in order to reduce the number of components that are necessary to build the predefined number of vehicles. For the presented method no predefined components are necessary.

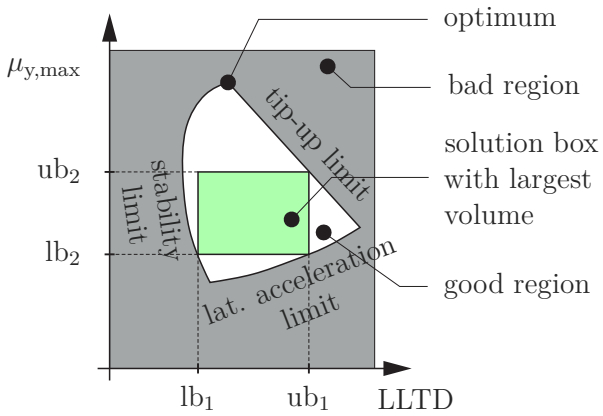


Figure 1.9: Solution space and solution box for the lateral load transfer distribution (LLTD) and the maximum grip of the tires ($\mu_{y,max}$) derived from vehicle dynamics requirements.

Conclusion. Solution Spaces represent a top-down methodology for deriving permissible intervals on design variables. Several techniques with different advantages and disadvantages were explored for their identification. If a valid quantitative model is available for the bottom-up computation and a robust design is required, solution spaces may be used to invert the problem. Solution spaces provide a practical and straightforward methodology for the design of systems.

1.2.3 Supervised Machine Learning

Machine learning is a wide field of study and can help the decision-making engineer to find better solutions in shorter times. It can be applied to any issue where it is necessary to make decisions for the future based on existing data from the past. Figure 1.10 gives

an overview of the field of machine learning divided into types of learning, categories of algorithms and techniques. In this work, neural networks are used to build regression models on one side, and *support vector machines* (SVM's) are used to classify data on the other side.

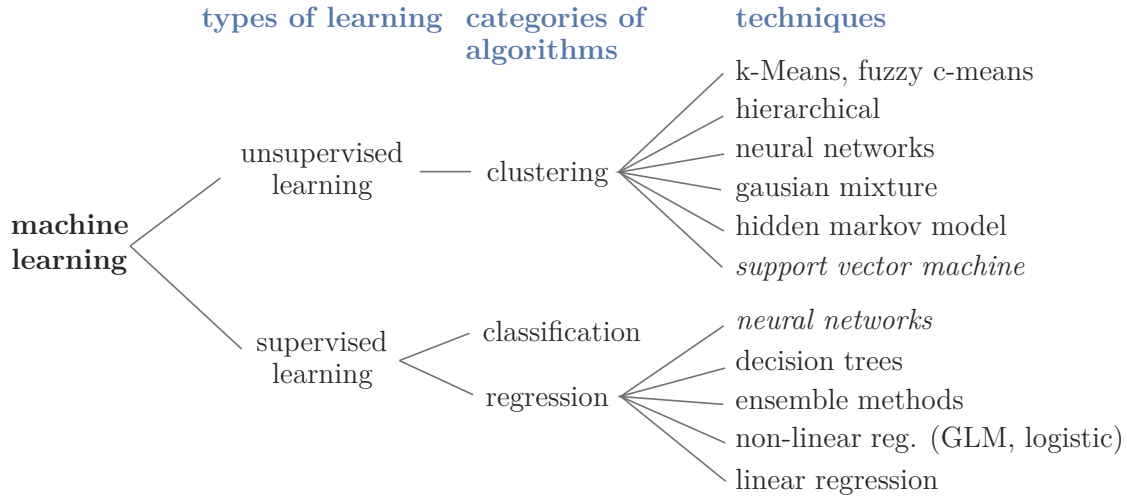


Figure 1.10: Machine learning – overview according to The MathWorks Inc. (2016). Techniques which are used in this work are written in italics.

Differences between clustering, classification, and regression. In the following, the main differences between different applications of machine learning are presented:

- *Clustering* regards the aim to subdivide a set of data into groups based on their properties. It is distinguished between problems where the number of clusters is prescribed by the user, and others, where the number of clusters results from the applied clustering technique. In the first case, the distance of an object to the closest cluster center decides which cluster the object belongs to. Therefore, the k-Means clustering technique is an iterative procedure where the objects are classified regarding all possible clusters before the center points of the clusters are recalculated. In the second case, the hierarchical clustering technique provides a binary tree that presents the number of clusters depending on the allowed distance between an object and the closest cluster. Therefore, the smaller the allowed distance, the more clusters exist. If the number of clusters is not prescribed by the user, the silhouette width criteria presented in (Rousseeuw 1987) can be used to define meaningful clusters. For more information about clustering techniques see (Duda, Hart, and Stork 2001).

  Example: A group of future vehicles is clustered by the required EPS-power and steering rack force in order to define relevant concept studies, see Figure 1.11 (a).

- *Classification* allows to predict a categorical response, the particular categories are prescribed by the user. In order to decide which category is assigned, the classification algorithm needs to be trained for which features the classification should be conducted. Therefore, a group of objects that have already been classified is required

in order to start the learning procedure. Multiple hyperplanes are used to reduce the number of misclassified objects for problems with many features.
Mapping: feature vector → class value

Example: Based on its properties, a vehicle must be evaluated as to whether it is good or bad, see Figure 1.11 (b). The property or feature of the vehicle, for example, can be the maximum lateral grip on the front and rear axle.

- *Regression* may be used for mapping relationships among variables in a quantitative manner.
  Mapping: feature vector → performance vector

  Example: The input to output relationship of a physical simulation model should be represented by a regression network, see Figure 1.11 (c).

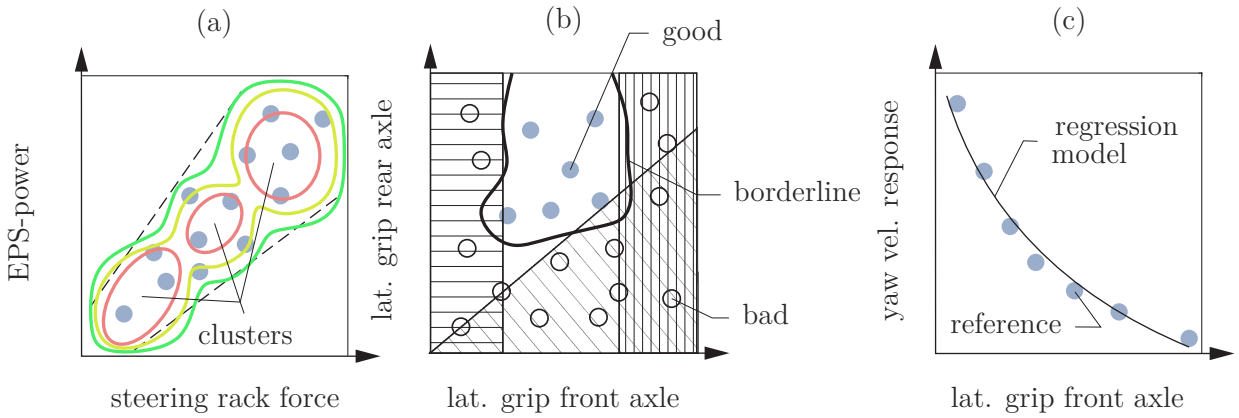


Figure 1.11: Machine learning – categories.

The following common steps need to be completed in order to obtain such an artificial meta-model based on machine learning: First, the data needs to be imported, cleaned and explored, second, the model is trained and tested, and third, if the model is considered good, it may be deployed.

The application of machine learning to the field of vehicle dynamics has increased continually over the past years. In (Chandrasiri, Nawa, and Ishii 2016) the support vector machine technique is used to classify the driving skills of different drivers. For more information about machine learning in general see (Michie, Spiegelhalter, and Taylor 1994).

Neural Networks

Neural networks are networks of nerve cells in the human and animal brain (Ertel 2013). Since artificial intelligence was applied to *artificial neural networks* (ANNs) by McCulloch and Pitts in 1943, their structure has been inspired by the biological brain (McCulloch and Pitts 1943). Therefore, they consist of neurons (nerve cells) that can be arranged parallel or sequentially. Neurons that are arranged parallel belong to the same layer. An informational exchange between neurons within the same layer is not allowed. Multiple

layers can be added behind each other in order to use the output of previous neurons as an input for the following neurons.

In this work, feedforward ANNs, also called multilayer perceptrons, are applied. The advantage of feedforward ANNs is that they are very well suited for mapping non-linear relationships. As the term feedforward implies, information is only exchanged in output direction, which means that feedback loops are not allowed at all. Typical for feedforward ANNs is that they contain at least one or even more hidden layers and their neurons are connected to the neurons in the neighboring layers. Therefore, neurons inside of a hidden layer have no connection outside of the ANN.

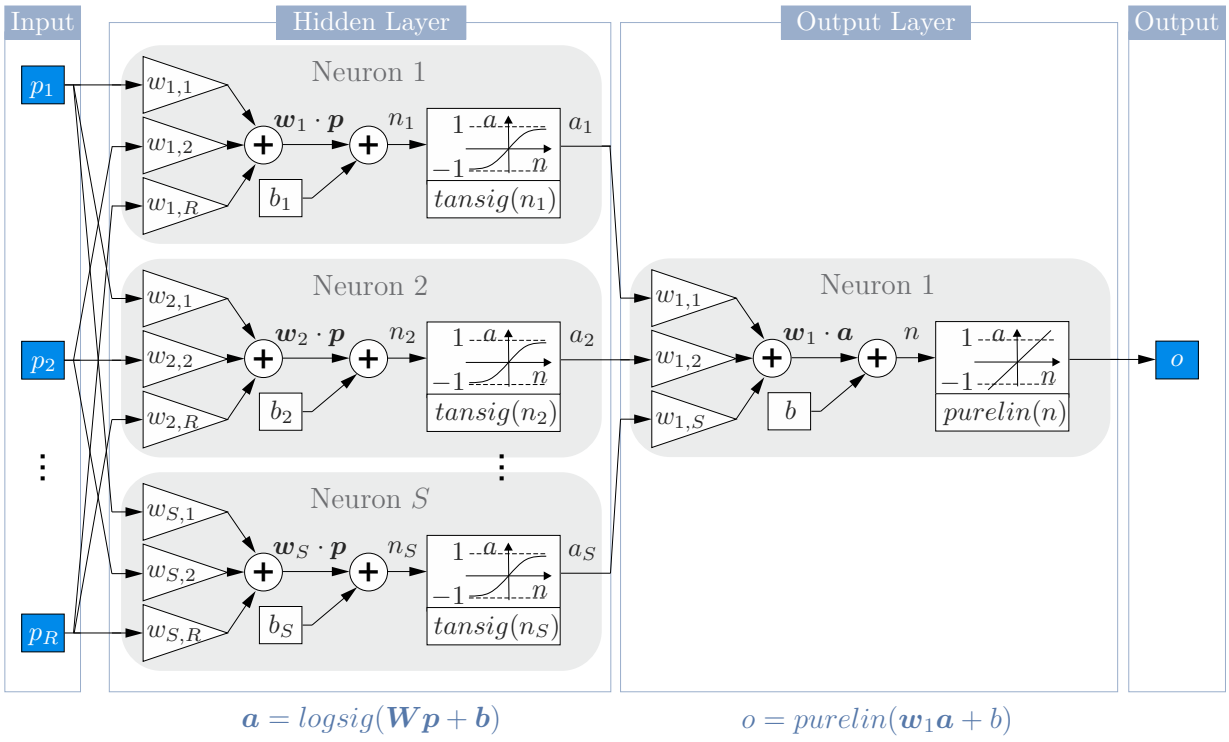


Figure 1.12: Structure of an artificial feedforward neural network.

In Figure 1.12, the structure of a feedforward ANN is presented. The ANN consists of R input variables p_i, one hidden layer which in turn contains S neurons and one output variable o. Each connection between a particular input variable and a neuron is multiplied by a separate weighting factor $w_{i,j}$, where i denotes the number of the neuron and j denotes the number of the input variable. In addition, within each neuron, a bias factor b_i is added. The resulting elements $n_1 \dots n_S$ are forwarded to the respective transfer function. Any differentiable function may be used as a transfer function to calculate the output of a neuron. However, usually log-sigmoid, tan-sigmoid or linear functions are used. As a result, the output of each neuron a_i is a function of the input variables, weighting factors, bias factor and transfer function and therefore lies in the range of -1 and 1 for the tan-sigmoid function and between 0 and 1 for the log-sigmoid. While the weighting factors influence the slope of a_i with varying input variables, the bias factor shifts the transfer function to the left or right. Within the output layer, the output of the previous hidden

layer is processed further in a similar way and scaled according to the requirements of the output variable. Since the presented structure offers numerous possibilities to connect the particular neurons, the connections are usually optimized automatically, while the number of hidden layers and neurons per layer is prescribed by the user. More detailed information about neural networks are available in (Michie, Spiegelhalter, and Taylor 1994), (Ao, Rieger, and Amouzegar 2010), (Grosan and Abraham 2011), (Krenker, Bester, and Kos 2011).

Support Vector Machine

Support Vector Machines are used for classifying data in this work. For a detailed view on their fundamentals see (Hastie, Tibshirani, and Friedman 2009; Cristianini and Shawe-Taylor 2000). Based on the given input data, it should be determined to which class a particular data point belongs. In the case of two available classes, this technique tries to find a hyperplane within the input space that separates the two such that first, the number of incorrectly assigned data points is minimized and second, the width around the hyperplane is maximized. The dimensionality of the hyperplane depends on the number of properties that are considered for each data point, also called features.

Figure 1.13 presents a random two-dimensional hyperplane $\mathcal{H}_0$ that classifies all the data correctly and an optimized one $\mathcal{H}_1$ with a maximized margin around the hyperplane. The bigger margin of hyperplane $\mathcal{H}_1$ results in less misclassifications if additional sample points should be classified, and therefore, $\mathcal{H}_1$ would be the preferable hyperplane.

The data points on the boundary of the hyperplane are also called support vectors as they support the position of the hyperplane. For SVM's, a few data points of the total quantity are enough in order to describe the hyperplane routing which makes this technique very efficient.

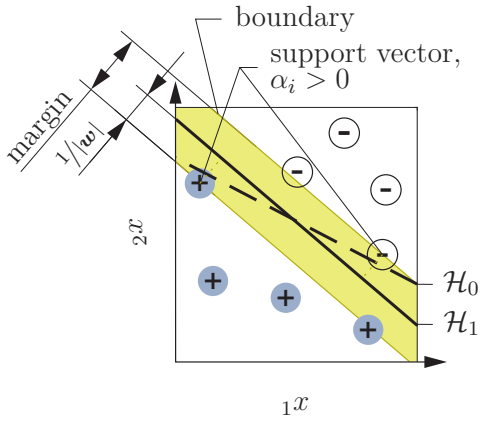


Figure 1.13: Linear Support Vector Machines.

Hyperplane. The class of a particular data point $\boldsymbol{x}$ may be measured by the objective criterion $y = \{-1,1\}$. Therefore, y is a function of $\boldsymbol{x}$ such that $y = \text{sgn}(\text{f}(\boldsymbol{x}))$. For multiple data points $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ the associated classes $\boldsymbol{y}$ are calculated by

$$\text{f}(\boldsymbol{X}) = \boldsymbol{w}\boldsymbol{X} + \boldsymbol{b} \tag{1.4}$$

with the bias $\boldsymbol{b}$ and the vector $\boldsymbol{w}$ which represents the width around the hyperplane to the support vectors (Müller et al. 2001). The input matrix $\boldsymbol{X}$ contains a sequence of data points and therefore includes all necessary information about the input data. On the other hand, $\boldsymbol{b}$ and $\boldsymbol{w}$ are unknown and usually defined by numerical optimization during the training procedure such that good classification performance is ensured. On the boundary

of the hyperplane depending on the side f($\boldsymbol{x}$) is equal to -1 or 1, on the hyperplane itself f($\boldsymbol{x}$) is equal to zero.

Lagrange Method. However, computationally it is more simple to extract $\boldsymbol{b}$ and $\boldsymbol{w}$ from the saddle points of the Lagrange function, which is equivalent to solving Equation (1.4)

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\,||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i \,(y_i \,(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) - 1), \tag{1.5}$$

where α_i is the Lagrange factor and i is the counting variable from 1 to the total number of training data points n; y_i represents the class of the associated input vector (Burges 1998). In order to calculate the saddle points, the first order derivatives of Equation (1.5) must be set to zero. By deriving and solving for $\boldsymbol{w}$, the equation $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, x_i$ is obtained. In addition, by deriving Equation (1.5) according to $\boldsymbol{b}$, we obtain $\sum_{i=1}^{n} \alpha_i \cdot y_i = 0$. Therefore, the Lagrange-multiplicators must be equal to or greater than zero $\alpha_i \geq 0$. This means that only for the support vectors alone the Lagrange-multiplicator α_i is greater than zero ($\alpha_i > 0$) and for all other data points, α_i is equal to zero ($\alpha_i = 0$). Since it is not possible to compute α_i directly based on the given equations, the following paragraph describes the required method necessary to do so.

Dual Problem. With the aid of numerical optimization, α_i is determined by solving the duality optimization problem described by Equation (1.6), where the index i and j associates particular data points such that all possible combinations are summed.

$$\underset{\alpha}{\text{minimize}}\; \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \text{ s.t. } \alpha_i \geq 0 \;\forall\; i \text{ and} \sum_{i=1}^{n} \alpha_i \cdot y_i = 0 \tag{1.6}$$

After α_i is computed, $\boldsymbol{w}$ can be calculated by $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, x_i$. In the following step, the bias may be calculated by $\boldsymbol{b} = \boldsymbol{y} - \boldsymbol{w}\boldsymbol{X}$, where $\boldsymbol{y}$ includes the classes of the training data. Since $\boldsymbol{w}$ and $\boldsymbol{b}$ are known, the resulting SVM can be applied in order to classify new data points by Equation (1.4).

Soft Margin. If it is not possible to classify the data without any misclassification, or if some error is expected within the assignment of the training data, a soft margin may be used. Therefore, Equation (1.7) must be minimized under the given constraints.

$$\underset{\boldsymbol{w}, b, \xi_i}{\text{minimize}}\; \frac{1}{2}\,||\boldsymbol{w}||^2 + C \sum_{i=1}^{n} \xi_i \text{ s.t. } \xi_i \geq 0 \;\forall\; i \text{ and } y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1 - \xi_i \tag{1.7}$$

The slack variable ξ_i indicates how far away from the hyperplane misclassifications can be expected for each class, see Figure 1.14. Since misclassifications should be avoided, ξ_i is minimized as well. Therefore, the penalty within the performance function is proportional to the distance between the respective boundary and the data point. The C-factor is provided by the user in order to adjust the fault tolerance. In the case of $C \to 0$, more misclassification is allowed and the margin is wider. However, if $C \to \infty$, the optimization

tries to find a hyperplane without any misclassification independent of the resulting margin. Due to the introduced slack variable ξ_i and the adopted boundary conditions, there is some empirical loss. Therefore it is not possible to identify the associated class of the training data based on the condition if $y_i > 0$ or $y_i < 0$ any longer.

Non-Linear Support Vector Machines. In "real world" classification problems, the particular classes often are not linearly separable, as presented in Figure 1.15. Since SVM's are only able to handle linear hyperplanes, the input space must be transformed into a linear separable space. Therefore, a so-called kernel function $k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$ is used to transform the input space S_{nonlin} to a higher dimensional linear separable input space S_{lin}. Following this, the support vectors are identified within S_{lin} in the same way as for linear separable data by minimizing Equation (1.8). The scalar product of two data points in the higher dimensional input space is equal to the one in the lower dimensional input space. Therefore, Φ does not need to be known as long as the kernel function is known.

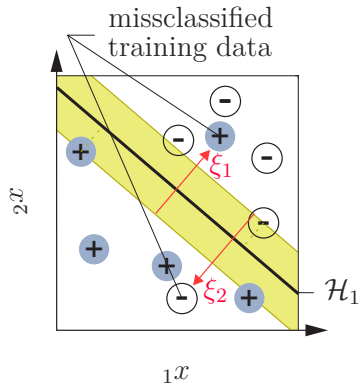


Figure 1.14: Support Vector Machines – soft margin.

$$\underset{\alpha}{\text{minimize}}\ \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\,\alpha_j\,y_i\,y_j(\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j))\ \text{s.t.}\ \alpha_i \geq 0\ \forall\, i\ \text{and}\ \sum_{i=1}^{n}\alpha_i\cdot y_i = 0 \tag{1.8}$$

In this work, the radial basis function (Gausian) is used as kernel function: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\frac{(-||\boldsymbol{x}_i-\boldsymbol{x}_j||)^2}{\gamma^2}$.

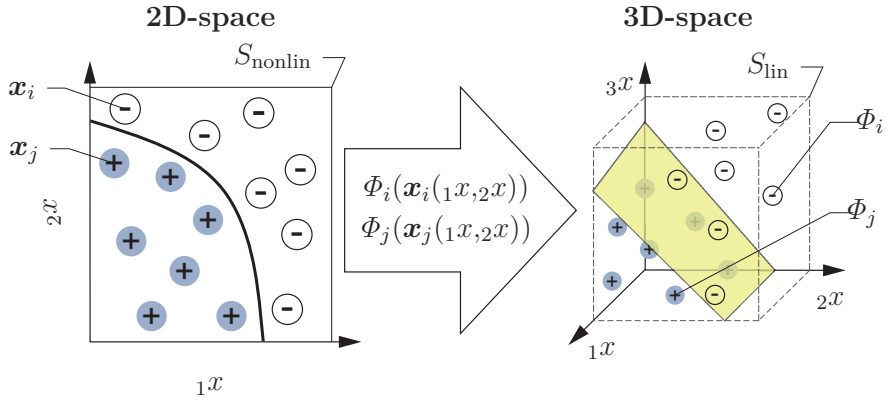


Figure 1.15: Non-linear Support Vector Machines.

1.3 Objective of the Work

The aim of this work is strongly motivated by the industry's economic wish to develop better vehicles in less time. Although the vehicle tends to become a more and more

networked, digitalized and autonomous device on four wheels, it is assumed that driving pleasure in the future still will be a selling point (Dudenhöffer 2016). However, since the focus of automobile manufacturers is shifting towards technologically more innovative fields, the design process of the classical systems (e.g., suspension, steering, tires) must be adapted in order to deliver high-quality products in less time. Therefore, the new method, which is developed in this work, needs to achieve the following aims.

Aim 1 *The first aim of this work is the development of a novel method for the design of Vehicle Architectures for Driving Dynamics. Therefore, a high number of design variables and objectives must be handled. In addition, non-linear system behavior needs to be treated in order to ensure the necessary accuracy. Due to the flexible nature of large and innovative projects, the method should cover a certain flexibility along the design process. Manufacturing inaccuracies may lead to design failure, therefore, the robustness of the final design against variations should be considered as well. In order to consider the whole set of all possible solutions, a top-down method is desired.*

A variety of research has been done regarding the top-down development of requirements in the field of vehicle dynamics (Wu et al. 2009; Kang et al. 2012; Eichstetter 2017). However, their scope is limited to certain subsystems and does not draw the whole picture of Architecture Design. Therefore, this work should enhance the scope of robust and flexible top-down development by a practical approach.

Aim 2 *In order to enable a fluent transition from Architecture Design in the early design phase to series development of the subsystems, the feasibility of the requirements derived on the subsystems should be ensured.*

When requirements on subsystems are derived by a top-down approach, their feasibility is not necessarily guaranteed. One reason for that is the limited level of detail of the underlying model during one particular top-down step. On one side, the complexity is limited since quantitative models with a higher degree of detail are often not available due to a lack of knowledge or due to the early design phase and, on the other side, detailed design models go hand in hand with an increased number of design variables which may no longer be practical above a certain number. For analysis of vehicle dynamics with respect to variations of tire properties, for example, some functional parameters of a model are tuned rather than the tread area or the rubber compound. In addition, interlinked design variables between subsystems of different departments complicate the solution identification due to individual feasibility restrictions. Therefore, the complexity of the applied models is and also needs to be limited.

Aim 3 *The design of a vehicle platform should be considered within the proposed method for the design of Vehicle Architectures. In order to reduce time-consuming iterations, common components for different vehicles should be considered on a detail-level with respect to the Vehicle Architecture design.*

Today, all series vehicle manufacturers develop and adopt platforms with respect to their future product portfolio. Therefore, a high demand for fast computer-aided methods that support their design in an efficient manner arises. In result, the customer may be provided with a high-quality product at lower costs due to economies of scale. Current design methods lack in practicality since they establish the possibility to share a particular component in different vehicles only on a functional level rather than on a detail level, which often is not enough for using the same component. In addition, considering the design problem piecewise in detail, but ignoring constraints to other relevant subsystems, may result in theoretical optimal solutions. Therefore, current methods do not consider the big picture, which often results in iterations and unnecessary compromises.

1.4 Concept of the Solution Approach

Successful products are built by combining different fields of technology and knowledge more and more often (Hab and Wagner 2012). Since customer expectations on successor products increase their complexity regularly, the complexity of the underlying processes increase as well if development processes are not adapted appropriately. Therefore, in this work, a new interlinked approach for the design of Vehicle Architectures for Vehicle Dynamics is presented. In order to cope with the challenges of developing the next generation premium vehicles, various demands on the development process arising from different requirements on the product must be devoted. In the 90's, researchers expected that "flexibility" is one of the key factors for success in product development, especially in unpredictable, rapidly changing environments (Bhattacharya, Krishnan, and Mahajan 1998). Since the combination of knowledge from different areas, e.g., mechanical and electrical, gets increasingly important, the need for transparency during the development process grows. In order to reduce resource expensive back-tracking in the development process during a progressing timeline, not only flexibility, but also establishing feasibility is key for success.

1.4.1 Requirement Management for Complex, Non-Linear Systems

A complex system may be characterized by its non-linearity, the number of design variables and objective criteria as well as the number of causal relationships between both. Non-linearity describes the shape of the causal relationships more in detail and may increase complexity. The development of complex, non-linear systems can be time-consuming and paved with obstacles.

In order to increase the efficiency in automobile development, it makes sense to structure the design procedure hierarchically. The hierarchical structure allows considering a reduced set of design variables in the early design phase and an extended one in the late design phase. This structure allows to concentrate on the relevant system properties first. Therefore, in this work, the vehicle is divided hierarchically into three abstraction levels: system level, subsystem level, and component/detail level. Between system level and subsystem level, customer-oriented requirements on functional properties of the relevant subsystems

are derived. The requirements on the subsystem level may be clustered regarding the considered subsystems in order to allow decentralized development by different departments. According to the specified functional requirements, the particular subsystem must be designed in detail. However, if a particular department is not able to design the subsystem according to its requirements, the requirements derived on other subsystems may have to be adapted in order to fulfill the top-level requirements. This mutual dependence usually slows down the entire design procedure.

How can this procedure be realized in engineering practice more efficiently? Based on the increased number of alternative solutions, it is easier to find a common position between different departments, if regions including multiple good designs are discussed rather than one particular design point that satisfies all the requirements. Therefore, solution spaces are used to invert the calculation of ordinary simulation models and break down the top-level requirements to the subsystem level (*Architecture Design*). Solution spaces provide the engineer with *flexibility* by offering several good designs and *robustness* versus unwanted design variations such as manufacturing tolerances. Due to solution spaces, robustness may be considered in the input space which makes it easy for the decision making engineer to find trade-offs, e.g., between robustness and all other requirements such as limitations of the modular system, costs, and the feasibility. In (Thomke 1997), the term "flexibility" refers to the possibility of adapting a design in response to a changing environment with little or no penalty, which is also possible within the solution space.

Novelty Value 1 *Development of a novel method for the design of Vehicle Architectures for Driving Dynamics that takes uncertainty and feasibility into account.*

In order to derive reliable and quantitative requirements for subsystem properties of complex, non-linear systems, detailed simulation models are necessary. In addition, the design of vehicle architectures for driving dynamics needs to consider a number of design variables and performance measures. For analytical solution space algorithms, the considered complexity (number of design variables and their design range, non-linearity etc.) is limited, therefore, a stochastic algorithm is used. A drawback of stochastic solution space algorithms is that they need a large number of simulations. Therefore, they can be time-consuming, which is critical in the early design phase. Due to Machine Learning, it is possible to reduce the calculation-time significantly, while the accuracy of the results suffers only slightly.

Novelty Value 2 *This work extends the framework considered by meta-models in vehicle layout design through an extended design space plus additional design variables compared to (Eichstetter 2017; Wimmler et al. 2016). Based on fast computing surrogate models including all interlinked design variables and their dependencies, the vehicle can be considered as a whole.*

1.4.2 Assurance of the Technical Feasibility of the Requirements

Technical feasibility of the requirements derived from the system on the subsystems is important in order to avoid time-consuming back-tracking. Consider the suspension system

of a vehicle. The characteristic curve of the scrub radius over the steering angle can be described with a polynomial of third order by prescribing the value of the scrub radius at minimum, zero, and maximum steering angle, see Figure 1.16. The scrub radius at zero steering angle is basically defined with respect to the vehicle behavior while braking in the corner, straight line behavior and wobbling of the tire. Therefore, the value usually lies within the range of $-10 \cdots +10\,\text{mm}$. Although it would be preferable to have a small scrub radius at the maximum and minimum steering angle in order to reduce the required steering support power of the electronic power steering (EPS), this is not always possible due to kinematic and geometrical limitations of the control links. Additionally, the shape of the curve is continuous, and the associated design parameters must stay in a certain relationship in order to be feasible. Therefore, arbitrary requirements on the *characteristic values* (CVs) of the scrub radius, which are purely based on top-level requirements, can lead to unfeasible requirements. To ensure the feasibility of the requirements derived on the subsystems, some information about the feasibility on the component level must be present on the subsystem level. Feasibility of requirements regarding the suspension system is examined by adapting the design variables on the component level by numerical optimization in order to identify a robust design. If no feasible design exists on the component level that also satisfies the subsystem requirements, the optimization results provide the engineer with possibilities to adapt particular requirements on the subsystem in order to improve the performance and robustness of the entire system. This feasibility optimization is one of the key factors for a consistent design across the different abstraction levels.

Novelty Value 3 *Feasibility restrictions are considered in the early development phase in order to increase transparency and reduce iteration loops.*

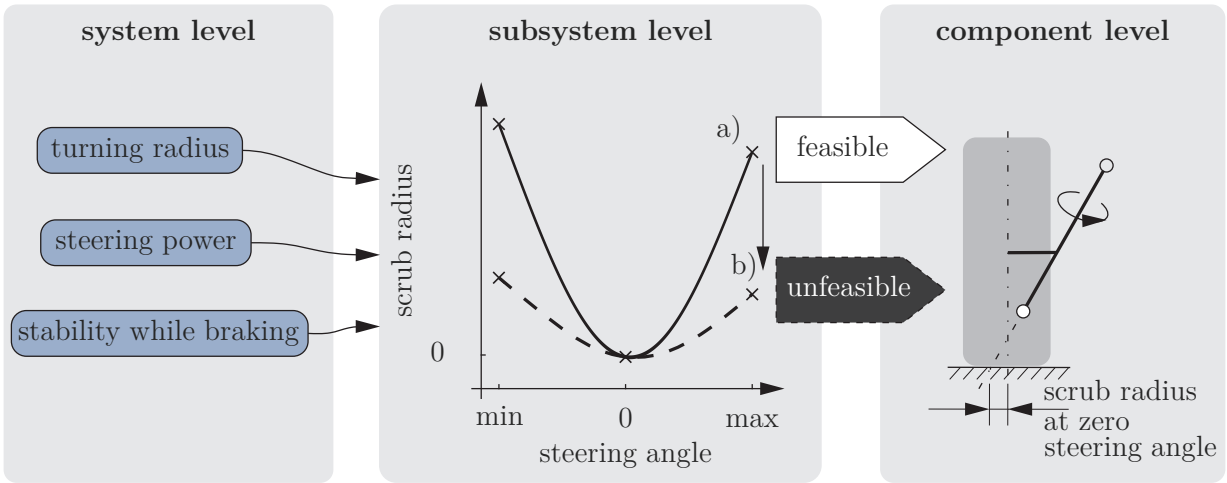


Figure 1.16: Feasibility issue of subsystem requirements on the example of the scrub radius. While characteristic curve b) might be the better choice in terms of system performance, characteristic curve a) also satisfies the system requirements and is feasible as well.

1.4.3 Consistency of Levels and Subsystem

Consistency plays an important role in the proposed development procedure for the improvement of transparency and efficiency. However, the term consistency is used in two different contexts in this work: consistency across the different abstraction levels and consistency of the subsystems or components.
Consistency across different abstraction levels designates the straightforward, top-down development process without undesired iteration loops and time-consuming back-tracking. As the design variables on the detail level are determined later in the development process, they somehow represent the timeline of the development process. In order to ensure this kind of consistency, all top-level requirements must be broken down across each abstraction level. Therefore, it is not permissible that a design variable from the bottom-level has a direct influence on the top-level by skipping the intermediate level.

Consistency of subsystems or components refers to the question of how good the elements on one particular abstraction level interact with each other. The underlying intention is that a better product may be designed with inferior components, if these components are tuned to each other. Consider two vehicles with a different lateral grip potential. Although the one with a higher lateral grip seems to be the better choice in terms of vehicle dynamics, this is not necessarily the case if the front axle does not fit to the rear axle, and thus causes the vehicle to become uncontrollable. Therefore, it needs to be ensured that all subsystems and components work well together and complement each other.

Since information that changes nothing is useless and information that changes a lot is very useful, relevant information is exchanged and integrated hierarchically across levels and horizontally across subsystems in order to ensure consistency.

Novelty Value 4 *Introduction of a consistent hierarchical structure for the design of vehicle architectures in the field of driving dynamics.*

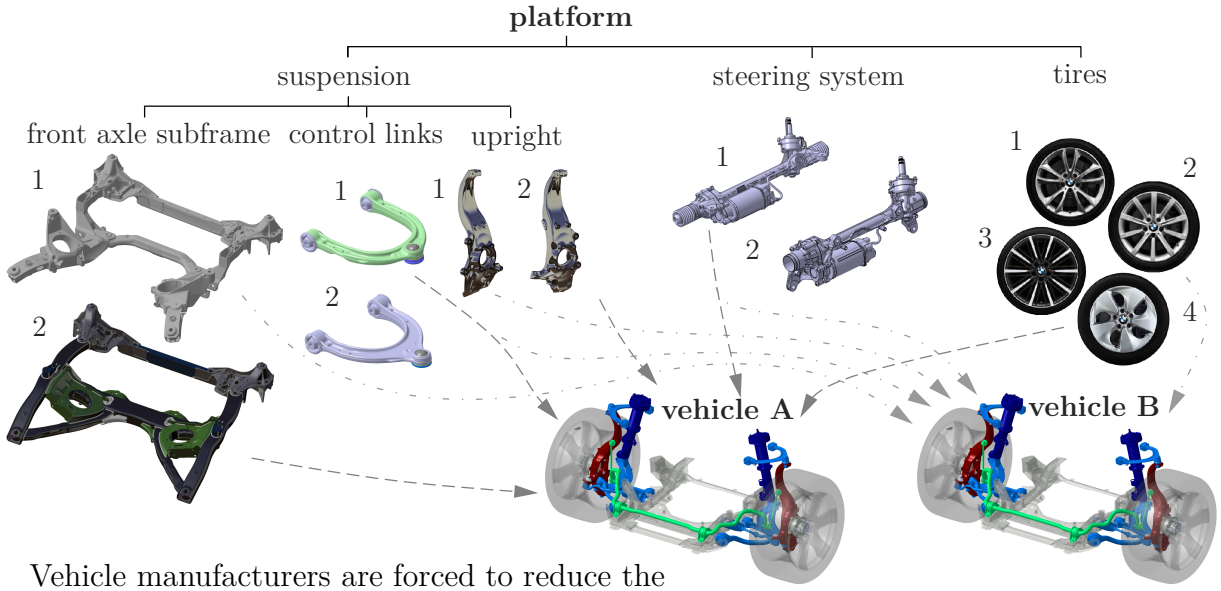
1.4.4 Architecture Design for Vehicle Platforms

Architecture Design for vehicle platforms allows different vehicles to share the same components or subsystems. Considering a particular vehicle platform already during the Architecture Design results in wider possibilities for adjusting other components in order to enable commonality and defining architecture interfaces. Therefore, interfaces between different components as well as functional interactions may be considered from the beginning on. Since those subsystems must perform properly in several systems, they have special requirements in terms of robustness and flexibility. Another challenge is to maintain the differentiation between the vehicles, especially of their top-level system properties which are directly related to the perception of the customer. By using vehicle platforms, the number of components may be reduced, which results in less effort for designing, integrating, and testing them. In this work, a systematic approach for developing a vehicle platform is proposed. By the combined use of solution spaces and multi-objective optimization across several abstraction-levels, a solution is achieved in a transparent way.

Definition 1 (Vehicle Architecture)
A vehicle architecture denotes the top-level system composition of various subsystems. Those subsystems may belong to a specific vehicle platform or not.

The decisions for a specific vehicle-DNA, tire diameter, suspension system or EPS-type are closely linked with vehicle architecture design. Although, decisions concerning a vehicle architecture are normally made based on system design analysis, they may need to be adapted, if they can not be realized for certain reasons. Therefore, the vehicle architecture provides a basic framework for the design of all other components.

Definition 2 (Vehicle Platform)
A vehicle platform consists of subsystems and components that are intended to be used on more than one vehicle. In addition, one vehicle platform can include various types of the same subsystem or component, for example, two different steering gears (left-hand/right-hand drive) that are used in various vehicles may belong to the same vehicle platform.



Vehicle manufacturers are forced to reduce the number of components associated with a product family in order to reduce costs. The challenge, meanwhile, is to maintain the individual character of each product. Although contemporary design methods using computer-aided approaches may result in the same overall performance as the previous product family despite a reduced number of components, the maximum potential performance rises by the use of an increasing number of components. Usually, the number of components is based on previous generations of platforms with slight adaptations. Newer approaches described in (Eichstetter, Müller, and Zimmermann 2015) use computer-aided techniques to

vehicle	A	B
subframe	1	2
control links	1	1
upright	2	1
steering system	1	1
tires	4	2

Figure 1.17: Example for a platform of vehicle dynamics relevant subsystems and components.

find the optimal number of components for a platform. Nevertheless, considering all the constraints, such as cost reduction per batch size and increase in difficulty computer-aided techniques often do not reflect the "real world". Therefore, in this work, it is assumed that the number of permitted variants of a component is defined. As a result the components must be designed such that all requirements on each system / vehicle of the product family are satisfied. Figure 1.17 shows an example of a vehicle platform containing components of the suspension/steering system and tires as well as two vehicles of the associated product family. While both vehicles use the same upper control link and steering system they have a different subframe, upright and wheel-set assigned from the platform. All other components do not belong to the platform and are designed specifically for each vehicle in order to satisfy the requirements on the particular vehicle.

Novelty Value 5 *Development of a novel method that designs Vehicle Architectures for Vehicle Platforms and considers commonality on a detail-level. Therefore, feasibility is not only ensured for the requirements on the subsystems, but also for the components within the modular system. As a result, several vehicles may share the same components or subsystems.*

1.5 Structure of the Work

In order to design a vehicle top-down along the V-model, this work is divided into five core parts. The first three treat different design problems and are presented in Figure 1.18. For all of the three subproblems, the same structured approach is used in order to achieve transparency in the design process. This structured approach includes three tools that are based on each other: *dependency graphs*, *quantitative models*, and *design algorithms*.

- *Dependency graphs* provide an intuitive overview of the relationships between design variables and objective criteria.
- *Quantitative models* provide the performance values of objective criteria based on a parameter set of design variables. In this work, physical and empirical models are used to quantify the relations between design variables and objective criteria. In addition, machine learning techniques are used to create RSMs and classifiers in order to increase the efficiency of the design algorithms.
- The *design algorithms*, finally, help to find an optimal solution in less time with respect to the considered goals and constraints.

In *Chapter 2*, a method for deriving requirements on large, complex systems is presented using the example of vehicle architecture design for driving dynamics. In order to ensure that the top-level requirements are achieved, the main influencing design variables from the field of, vehicle-DNA, tires, suspension, and steering are identified and considered. Due to the complexity of the problem plus the required flexibility and robustness, solution spaces are used to identify possible solutions. To increase efficiency neural networks are used for computing solution spaces.

Chapter 3 and *Chapter 4* relate to the feasibility of the former derived requirements by finding particular solutions on the detail level. The therefore proposed methods can be considered as independent design approaches that solve particular design subproblems. For the design of the tire, based on machine learning techniques, a novel quantitative model is created. The novel tire model represents the relationships between FTCs and full-vehicle properties. In addition, a classifier identifies unfeasible regions within the design space. Both models may be overlaid and then used to find feasible solution spaces for FTCs. Regarding the suspension design, a robust multi-objective optimization algorithm is applied between the subsystem and detail level. Therefore, the solution box on the subsystem-level is further processed to a particular solution on the detail-level.

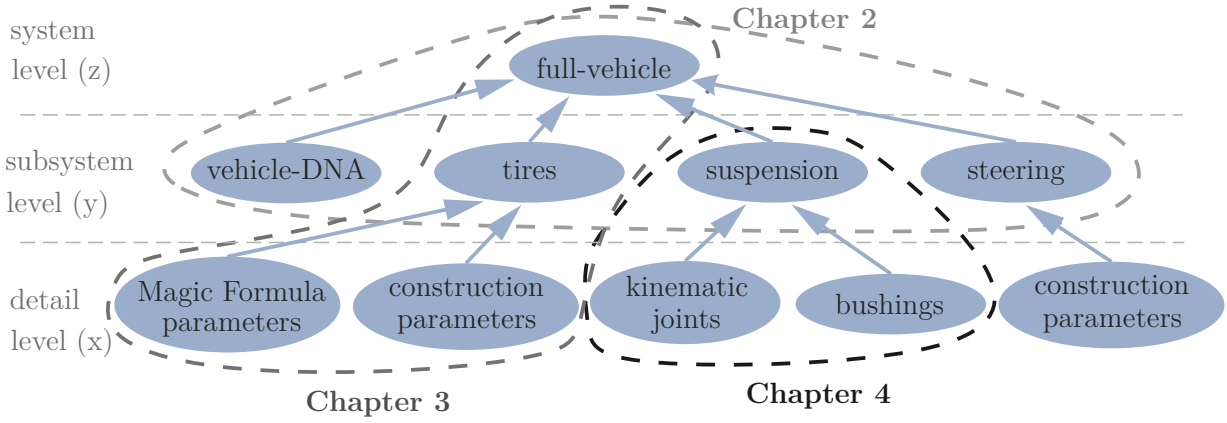


Figure 1.18: Main subjects of this thesis at a glance.

In *Chapter 5* two interlinked design approaches that enable the consistent design between the particular abstraction levels (system-, subsystem-, detail-level) are presented. Therefore, top-level requirements are derived top-down with respect to feasibility restrictions. It is distinguished between an intuitive semi-manual and a fully automated algorithm. Both of the presented algorithms may be applied to particular vehicles as well as vehicle platforms.

Chapter 6 relates to the application of the presented design methods. Therefore, contemporary, practical design problems of vehicle manufacturers are solved by the use of the presented methods.

In *Chapter 7* the results are discussed, followed by *Chapter 8* in which conclusions are made and an outlook is given.

CHAPTER 2

ARCHITECTURE DESIGN IN VEHICLE DYNAMICS

Since the emphasis of this work relies on the design of vehicle dynamics, all contributing properties of the following subsystems must be considered: vehicle-DNA, tires, suspension, and steering. Figure 2.1 presents the scope of the architecture design treated in this chapter. In the architecture design, requirements on particular subsystems are derived based on objectified system level requirements. According to the subsystem requirements, the specific components are developed, see Chapters 3 and 4. As the architecture design provides the only possibility to balance the requirements between the different subsystems, it provides the framework for potential customer satisfaction as well as the necessary effort for satisfying the subsystem requirements.

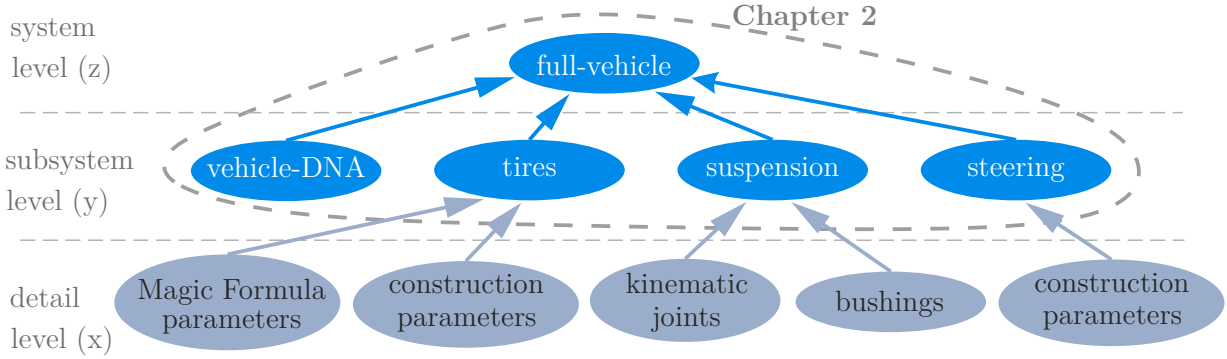
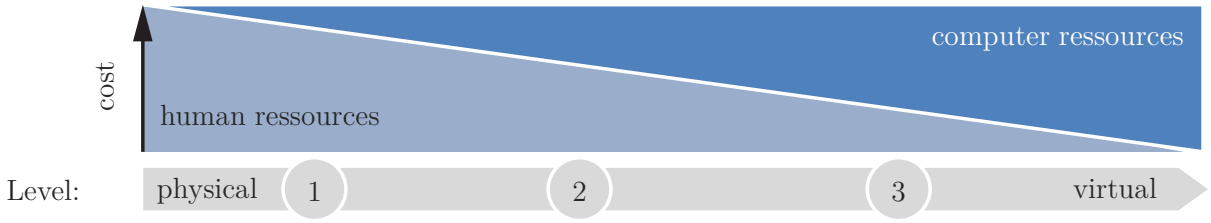


Figure 2.1: Architecture design as part of the interlinked design procedure.

The basic idea behind this novel method for *architecture design in vehicle dynamics* is an *interlinked design* method for the design of a consistent product in an efficient manner similar to Kim (2001). It relates to the virtual vehicle development procedure. In this context consistent means that all components are tuned to each other to achieve the desired

system performance. Efficient means that cost expensive iteration loops in later development phases are reduced to a minimum. The foundation of the novel design approach presented in this work is based on three tools similar to Zimmermann et al. (2017):

- *Dependency graphs* that include the scope of design variables and objective criteria as well as the knowledge about the qualitative relationships between them, see Section 2.1.
- *Quantitative models* that evaluate the performance of the system realistically enough in order to make major design decisions based on them, see Section 2.2.
- *Design algorithms* to find an appropriate solution, e.g., numerical optimization or solution spaces, see Section 2.3.



Enabler:	**Dependency Graphs**	**Quantitative Models**	**Design Algorithms**
Design process:	trial and error with hardware modifications based on qualitative knowledge	simulation-based trial and error based quantitative relations	definition of target design values (point based design) or solution intervals (solution spaces) combined with a systematic approach to find the optimal solution
Evaluation:	by multiple drivers; subjective deviation between desired and present behavior	deviation between desired and present objective parameters	classical point based optmization: deviation between desired and present objective parameters; solution spaces: vehicle satisfies the requirements or not

Figure 2.2: Foundations for a consistent and efficient design approach.

All three tools build on each other. Therefore, the scope of design variables and objective criteria needs to be defined first. In addition, the qualitative relationships between design variables and objective criteria need to be identified and documented in an appropriate form, such as a *dependency graph*. Based on the qualitative knowledge about the relationships between the design variables and objective criteria, a quantitative model may be

created in order to virtually asses the performance of a particular design. However, the accuracy of the results produced by the *quantitative model* depends on the considered design variables and the degree of detail pertaining to their impact on the system response. Vice versa, a dependency graph may also be derived by sensitivity analysis based on an existing simulation model. The quantitative model may be implemented into a *design algorithm* in order to find a solution that satisfies the objective criteria in an efficient manner. As a result, not every design needs to be evaluated by a human, since the design algorithm performs this task. Figure 2.2 compares the three tools in terms of cost, design process and evaluation of the results.

As a result, the real design work is done by the design algorithm, which mainly defines how the product needs to be produced. However, since the designs proposed by a design algorithm do not only depend on the performance functions and considered constraints, but also on the accuracy of the simulation results, the applicability of the method depends on each of the three tools.

According to the functional requirements derived on the subsystem level during the architecture design step, the particular subsystems presented in Figure 2.3 are designed.

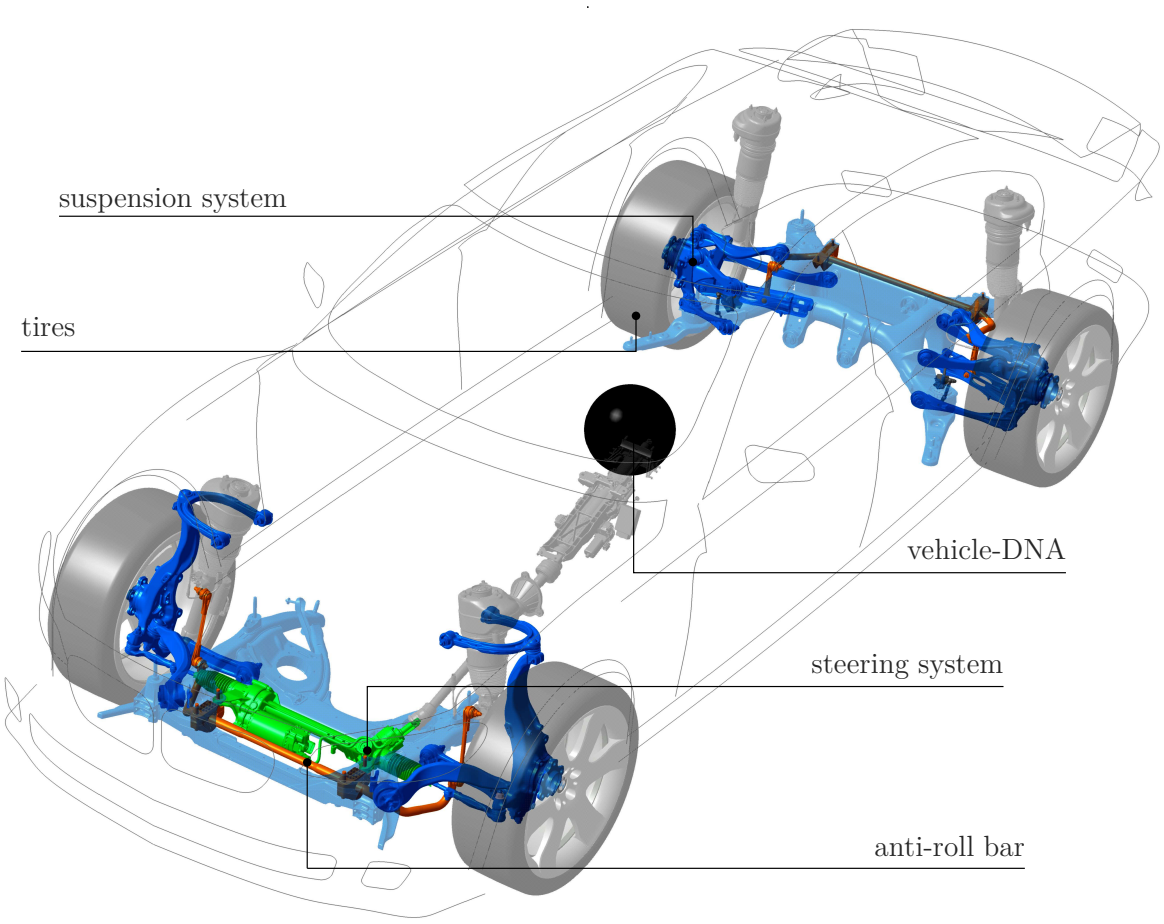


Figure 2.3: Subsystems considered for the design of vehicle dynamics.

2.1 Dependency Graphs

Dependency Graphs provide information about dependencies between objective criteria and design variables in a qualitative and graphical way on several abstraction levels. Ideally, dependency graphs are based on "real world" relationships, yet, if calculation models do already exist, they can also be created based on sensitivity analysis[4]. Therefore, a critical threshold value for the sensitivity measure is defined. If the sensitivity between a particular design variable and objective criterion exceeds a critical threshold value, a connection between both is drawn in the dependency graph. Especially in complex systems, they are helpful in order to identify possible key-parameters for tuning the system performance, robustness, and quality.

Rules for Dependency Graphs treated in this work:

- Back propagation is not allowed
- Loops are not allowed
- All parameters on the same level are independent of each other

In order to improve transparency and provide better insight into the causal relationships, dependency graphs may be divided into different abstraction levels. Therefore, in this work, the properties of the vehicle are divided into three hierarchical levels: system, subsystem, and detail level. In addition, it is useful to select the parameterization of the objective criteria and design variables such that cross-links between them, and therefore complexity, is reduced. As a result, larger solution boxes may be calculated which are described in Section 2.3.

The level of detail shown by the dependency graph may vary depending on the demands of the user. Therefore, only causal relationships that have an influence above a certain threshold value may be visualized. Another possibility is to cluster certain properties that belong to the same abstraction level of the vehicle. In this work, the following three layers of detail are used to cluster the vehicle properties:

- coarse layer: properties are collected to big clusters
- medium layer: some properties are clustered
- detailed layer: none of the properties are clustered

The dependency graph in Figure 2.4 shows the causal relationships that are crucial for the vehicle architecture design on the coarse layer. The presented graph is created based on an existing map-based simulation model of a vehicle which will be introduced in Section 2.2. It computes the performance on the system level based on the input data on the subsystem level. Since the applied vehicle model is quite complex, a global

[4]For consideration of the whole design space global sensitivity analysis which can be distinguished in regression-based (e.g., *Pearson Correlation*) and variance-based (e.g., *Sobol Indices*) methods are preferred (Mastinu, Gobbi, and Miano 2006)

sensitivity analysis is performed to rank the importance of particular design variables. Therefore, a superset of design variables is varied in a *design of experiment* (DoE) using Monte Carlo sampling before the *Pearson's Linear Correlation Coefficient* ρ is computed for each combination of design variable and objective criterion. According to the sensitivity measure $|\rho^{zi}_{yj}|$ between the j-th input variable on the y-level and the i-th output variable on the z-level, the relationship between both is evaluated according to Equation (2.1).

$$s^{zi}_{yj} = \begin{cases} high & \text{if } 0.5 < |\rho^{zi}_{yj}| \leq 1.0 \\ mid & \text{if } 0.1 < |\rho^{zi}_{yj}| \leq 0.5 \\ low & \text{if } 0.0 < |\rho^{zi}_{yj}| \leq 0.1 \end{cases} \tag{2.1}$$

While relationships with a *low* sensitivity are excluded from the dependency graph, those with a *mid* sensitivity are represented by a thin line, and those with a *high* sensitivity are represented by a thick line. As the properties are clustered on the coarse view, a relationship between two clusters is shown as a thick line, if at least one particular relationship between any of the properties inside of both clusters is *high*. A thin line implies that at least one design variable of a cluster impacts at least one objective criterion of another cluster. The colors of the relationships refer to the associated cluster on the y-level. It is obvious that the properties of each subsystem affect multiple clusters of requirements, which reflects the high complexity of the design task. Additionally, it can be seen that all the subsystems are evaluated according to the required system properties on the highest level, which reflects the hierarchical structure of the design task.

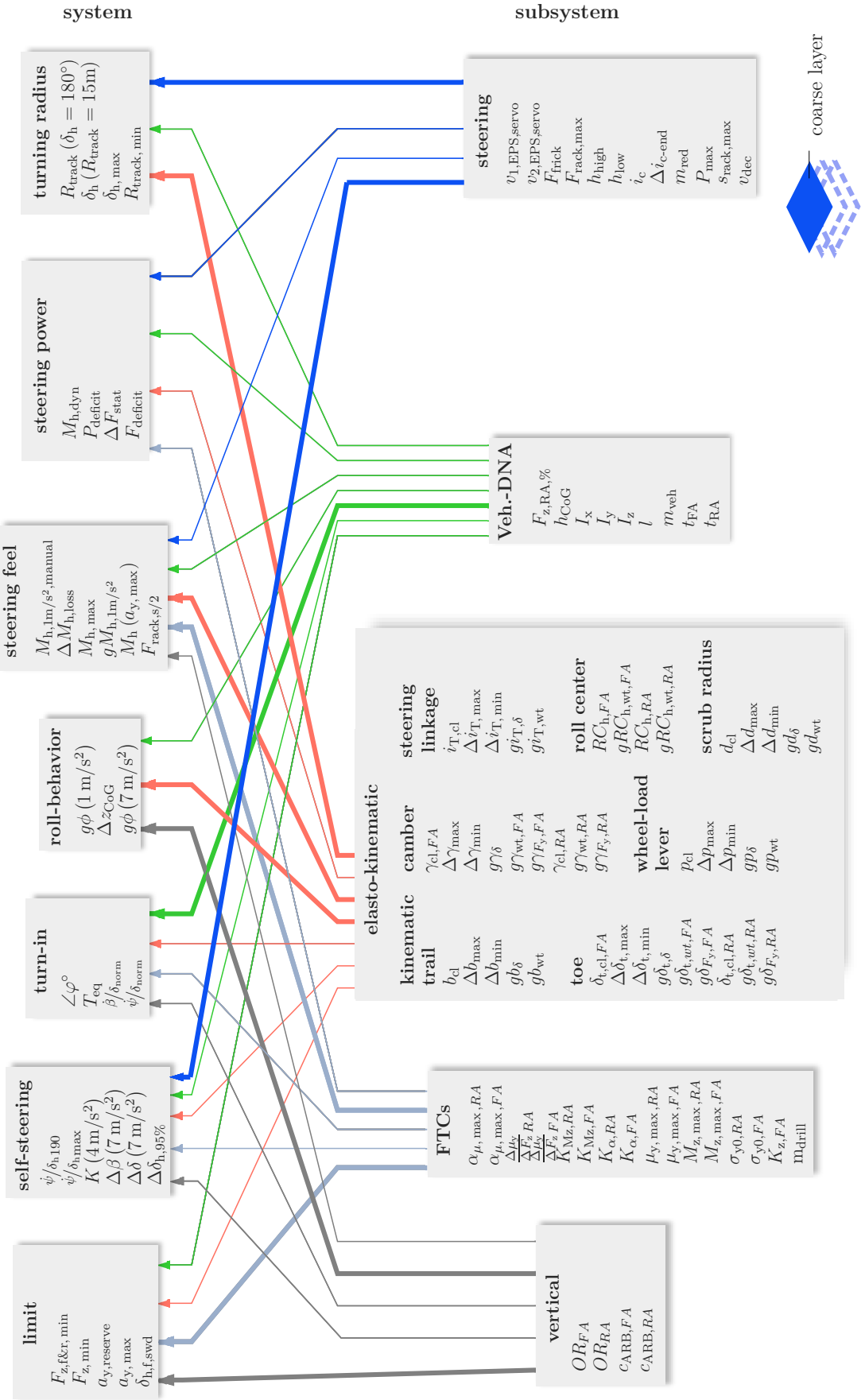


Figure 2.4: Dependency graph of the architecture design.

Figure 2.5 shows an excerpt of the dependency graph presented in Figure 2.4 on the detail layer. Therefore, all layout design relevant causal relations of the roll center height $RC_{\mathrm{h},FA}$ and its gradient over wheel travel $gRC_{\mathrm{h,wt},FA}$ are shown for the front axle. The circled number in the right upper corner of the objective criteria indicates the number of other design variables that influence the particular design objective as well. A dashed line represents a relationship with a low sensitivity. Since the jacking of the vehicle Δz_{CoG} is mainly dependent on $gRC_{\mathrm{h,wt},FA}$, but $RC_{\mathrm{h},FA}$ has a considerable influence as well, both design variables need to be considered during the design procedure. In contrast, for the design of the roll gradient at different lateral accelerations $g\phi\,(1\,\mathrm{m/s^2})$, $g\phi\,(7\,\mathrm{m/s^2})$ as well as the stability reserve $a_{\mathrm{y,reserve}}$, the gradient of the roll center height may be neglected, as its influence is too low to be relevant for the layout design of the vehicle.

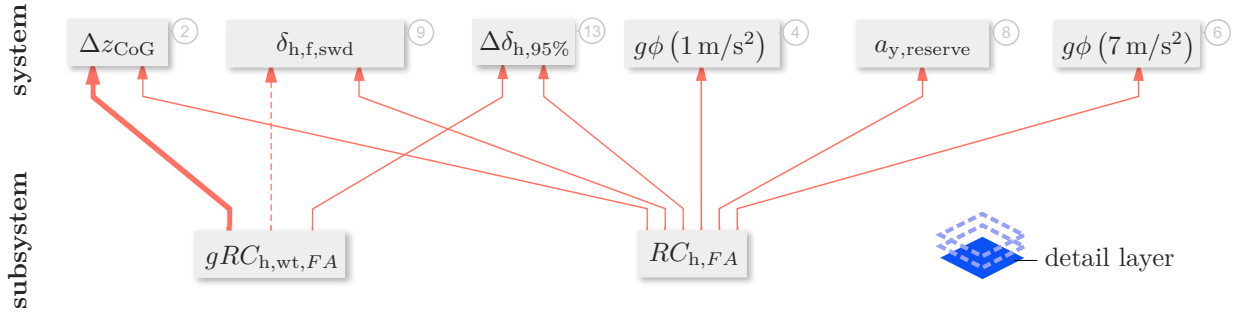


Figure 2.5: Dependency graph of the roll center height.

Another excerpt from the dependency graph presented in Figure 2.4 is shown in Figure 2.6. While the more general Figure 2.4 leads to the assumption that it is necessary to design all of the subsystems simultaneously, which often is a problem in engineering practice for the reason of uncertainties. Figure 2.6 shows that, depending on particular objective criteria, clusters of relevant design variables may be identified, while design variables of different subsystems belong to a cluster. Therefore, the maximum lateral grip coefficient $\mu_{\mathrm{y,max},RA}$ and cornering stiffness $K_{\alpha,RA}$ of the tire need to be tuned together with the gradient of toe variation over lateral force $g\delta_{F_{\mathrm{y}},RA}$ of the suspension, while the $g\delta_{F_{\mathrm{y}},RA}$, $g\delta_{F_{\mathrm{y}},FA}$ and steering linkage transmission ratio $i_{\mathrm{T,cl}}$ of the suspension need to be tuned together with the pinion ratio i_{c} of the steering system.

Statement 1 (requirement derivation)
While deriving requirements on design variables top-down, all design variables with an impact on the particular design objective need to be considered simultaneously to satisfy the requirements.

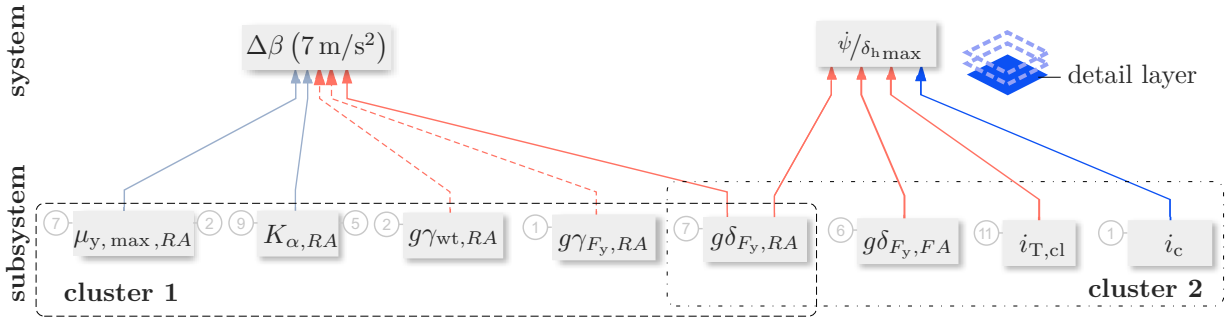


Figure 2.6: Dependency graph of the change in body slip angle $\Delta\beta\,(7\,\mathrm{m/s^2})$ and yaw velocity response $\dot{\psi}/_{\delta_{\mathrm{h}}\max}$.

2.2 Quantitative Models

Based on known qualitative causal relationships of the real system, quantitative simulation models can be created. Therefore, different modeling techniques may be used. In general, quantitative models can be distinguished between the following three main types:

- *Experimental* models are hardware build models that can be scaled and made of another material and afterwards tested under almost real world conditions, e.g., wind tunnel model at a scale of 1 : 2.
- *Physical* models are based on fundamental natural relationships that are combined with each other in order to build complex models. An example is the Hooke's law: $F = k\,X$.
- *Mathematical* models are based on mathematical formulas tailored with the objective to map a specified behavior between input and output variables, e.g., ANNs, linear approximations.

Those quantitative models may be also used for gathering more information about system behavior by varying the input parameters or to support design decisions. In (Guiggiani 2018; Schramm, Hiller, and Bardini 2018) the quantitative relations regarding vehicle dynamics are described in detail. More advanced design algorithms, presented in Section 2.3 for example, may help to generate better design decisions in a more efficient way.

2.2.1 Overview of the applied Quantitative Models

In this work, several models are used to quantify the causal relationships among the different abstraction levels. Figure 2.7 presents a rough overview of the models used and built in this work. Please note that the named models can be replaced by other types, e.g., a model which is based on the principles of physics can be replaced by an experimental one whose origin lies in the varied input and measured output data. In this work, the following models are used to quantify the relations between design variables and objective criteria.

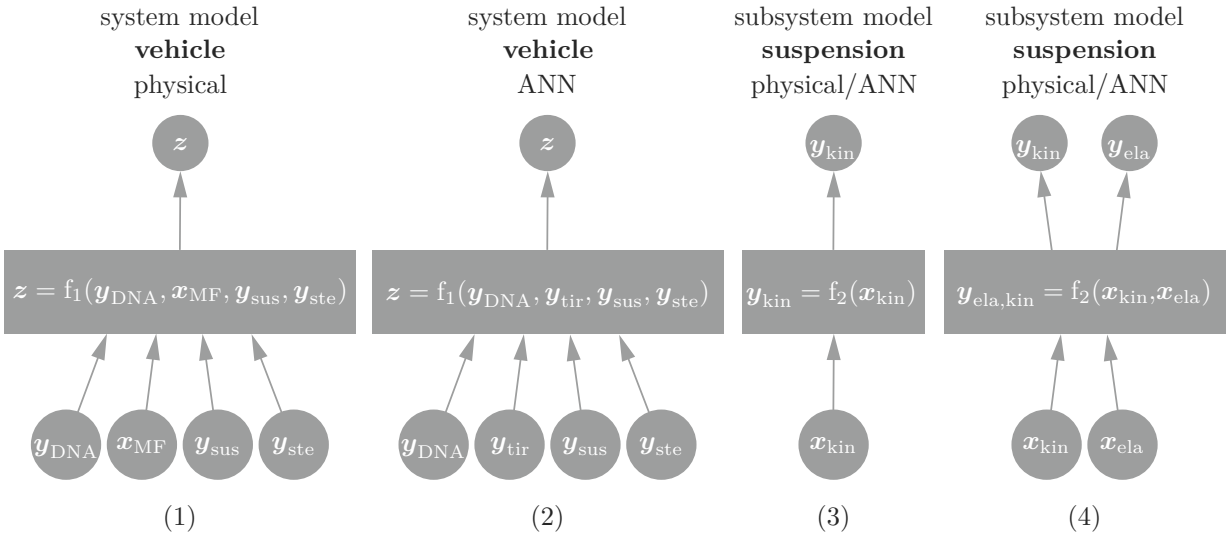


Figure 2.7: Overview of the simulation models used in this work: (1) a physical vehicle model, (2) a mathematical vehicle model represented as ANN processing the FTCs on the y-level as an input, (3) a kinematic suspension model represented as physical model as well as ANN, (4) an elasto-kinematic suspension model represented as physical model as well as ANN.

- The *vehicle/system* behavior is represented by a complex map-based two-track model that computes the system performance based on the input data on the subsystem level. The model itself is physically modeled and validated by multiple analysis of the BMW Group, which is why it is assumed to be accurate enough for the area of interest. Since the performance of the vehicle is evaluated in terms of its dynamic behavior and parking, only those subsystem properties that impact the associated objectives need to be considered as design variables, although the model has many more input variables. Design variables that are not considered for the Architecture Design in Vehicle Dynamics may be adopted from a similar vehicle. The relevant design variables may be clustered according to their particular subsystem, e.g., vehicle-DNA, tires, suspension and steering as shown in Figure 2.7 (1) for the physical model. For computing the system performance $\boldsymbol{z}$, the physical vehicle model needs the properties on the subsystem level as input, which are: vehicle-DNA $\boldsymbol{y}_{\mathrm{DNA}}$, suspension $\boldsymbol{y}_{\mathrm{sus}}$ and steering $\boldsymbol{y}_{\mathrm{ste}}$ properties. The tire properties need to be taken from the detail level $\boldsymbol{x}_{\mathrm{MF}}$, as the applied MF-tire model is an empirical model that can not be parametrized by *functional tire characteristics* (FTCs). For decreasing computational effort, an ANN is created based on the physical model as described in Subsection 2.2.3. The input/output structure of this artificial vehicle model is presented in Figure 2.7 (2). In contrast to the physical vehicle model, the *artificial vehicle model* allows physical tire properties $\boldsymbol{y}_{\mathrm{tir}}$ as input from the subsystem level, the underlying methodology is described in Chapter 3.
- The subsystem behavior of the *suspension* may be represented by a kinematic (see Figure 2.7 (3)) or elasto-kinematic (see Figure 2.7 (4)) model. Depending on which

model is used, statements about the kinematic behavior $\boldsymbol{y}_{\text{kin}}$ or even elasto-kinematic behavior $\boldsymbol{y}_{\text{ela}}$ of the suspension can be made. Since the input of the vehicle model is always the complete set of suspension properties, which is defined by $\boldsymbol{y}_{\text{sus}} = [\boldsymbol{y}_{\text{kin}}, \boldsymbol{y}_{\text{ela}}]$, some elasticity properties need to be assumed, if a pure kinematic suspension model is used as subsystem model. However, in Chapter 5 it will be described why using a kinematic suspension model offers multiple advantages compared to a elasto-kinematic one.

- In the physical vehicle model, the subsystem properties of the *tire* are represented by the empirical MF-tire model. Using the *Magic Formula* (MF) within a requirement based top-down process has several disadvantages. The amount of MF parameters is large, and they can not be forwarded to a tire manufacturer in order to build a tire. Therefore, machine learning is used in order to replace the MF parameters with physical/functional properties of the tire, which are also called FTCs, see Chapter 3.

2.2.2 Physical Vehicle Model

The vehicle model is a map-based two-track model including 15 DoF. The DoF are the steering angle, the three rotational and translational movements of the vehicle body plus the wheel travel and rotational movement on all four wheels. Based on the maneuver which is performed as well as the drive concept, some of the mentioned DoF act as input variables. Therefore, the steering angle and the four wheel rotations (in the case of an all-wheel-drive) are no longer DoF which reduces their number to 10.

Parametrization of the physical vehicle model

In order to break down requirements on properties of different subsystems, they must be represented as scalar values. In the following, the design variables considered in this work are introduced separately for each of the four subsystems.

Vehicle DNA The vehicle-DNA comprises all basic properties of the vehicle. An overview of the properties considered in this work is presented in Table 2.1. For the presented design variables, the impact on the objective criteria is quantified, which serves the input data for creating the artificial vehicle model.

Table 2.1: Overview of design variables belonging to the Vehicle DNA.

design variable	**symbol**	**unit**
vehicle mass	m_{veh}	kg
track width front axle	t_{FA}	m
track width rear axle	t_{RA}	m
center of gravity height of the vehicle	h_{CoG}	m

Continued on next page

Table 2.1 – continued from previous page.

design variable	**symbol**	**unit**
wheelbase of the vehicle	l	m
static longitudinal wheel load distribution rear to front	$F_{\mathrm{z,RA,\%}}$	-
roll inertia of the vehicle	I_{x}	$\mathrm{kg\,m^2}$
pitch inertia of the vehicle	I_{y}	$\mathrm{kg\,m^2}$
yaw inertia of the vehicle	I_{z}	$\mathrm{kg\,m^2}$

Suspension Table 2.2 gives an overview of the parameters investigated in this work for the suspension design. Parameters that are indexed by "k/e" may be considered by a pure kinematic or by an elasto-kinematic model as well. However, parameters that are only indexed by "e" can only be considered correct by the use of an elasto-kinematic model. The gray filled fields indicate parameters that are only considered for the design of the front suspension. Therefore, the forces acting on the toe rod of the rear suspension are not investigated in this work.

Table 2.2: Overview of functional suspension properties on the y-level.

Functional suspension properties	Value at construction level	Gradient vs. wheel travel	Variation until min. steering angle	Variation until max. steering angle	Gradient vs. steering angle	Gradient vs. lateral force
toe (min)	k/e	k/e				e
camber (min)	k/e	k/e	k/e	k/e		e
scrub radius (mm)	k/e	k/e	k/e	k/e		
kinematic trail (mm)	k/e	k/e	k/e	k/e	k/e	
steering linkage transmission ratio (mm/°)	k/e		k/e	k/e		
roll center height (mm)	k/e	k/e				
wheel-load lever (mm)	k/e	k/e	k/e	k/e	k/e	

In a particular driving situation, the values of some of the functional suspension properties are dependent on the three variables: wheel travel, steering wheel angle, and horizontal forces on the contact patch of the tire. The contributions from those three variables to the value of a particular functional suspension property are comprised by the superposition method. Therefore, based on the parameterization shown in Table 2.2, several characteristic curves and maps are built in order to interpolate the values of the functional suspension properties for a particular driving situation. In Figure 2.8, different parameterizations for characteristic maps and curves are presented and compared with the measured ones.

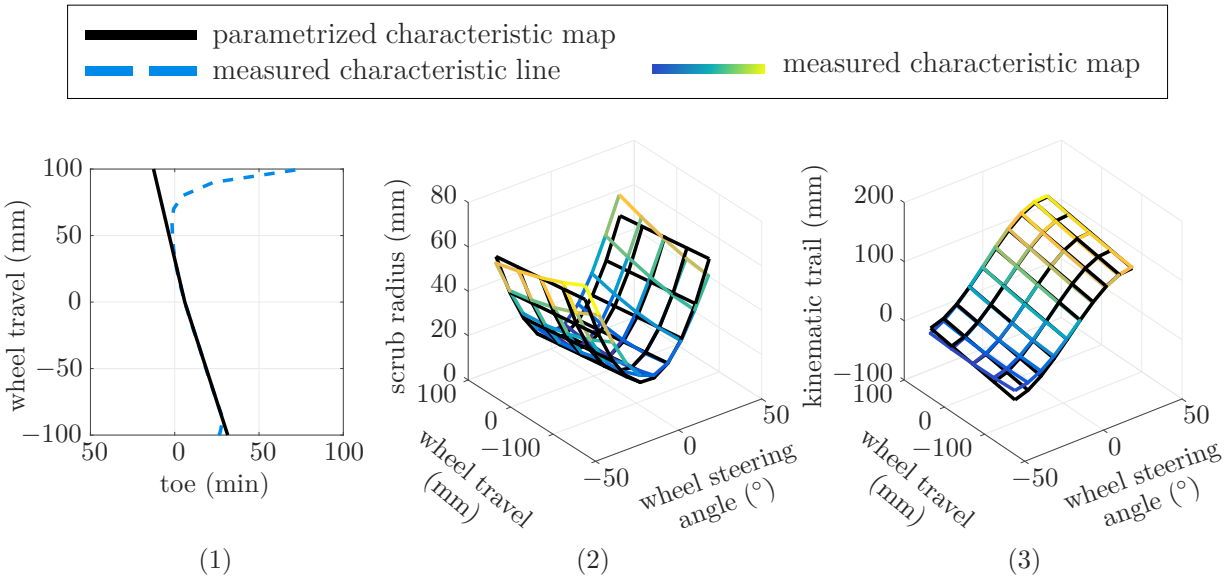


Figure 2.8: Comparison of measured and parametrized characteristic maps, (1) characteristic curve based on a first-order polynomial, (2) characteristic map-based on one gradient and a polynomial of second order which in turn consists of three values, (3) characteristic map-based on one gradient and a polynomial of third order, which in turn consists of three values and one gradient.

According to Table 2.2, the toe variation of a kinematic model may be described depending on the wheel travel. Therefore, the associated characteristic curve is described by the absolute value at construction level and the gradient along the wheel travel, see Figure 2.8 (1). The scrub radius, however, depends on the wheel steering angle as well. Therefore, a second order polynomial is used in order to represent the relations between the scrub radius and wheel steering angle, see Figure 2.8 (2). In addition, Figure 2.8 (3) presents the characteristic map of the kinematic trail. Since the gradient of the kinematic trail along the wheel steering angle is also important to generate a realistic map at construction level, a third order polynomial is used in order to describe the kinematic trail as a function of the wheel steering angle.

The accuracy of the results must be sufficient enough to apply the introduced parametrization in the design process. Therefore, the system response of the following two vehicle models is compared: The first model consists of exact characteristic maps as they were measured by the test rig, while the second model consists of maps generated based on functional suspension properties. The functional suspension properties of the second vehicle model were identified and extracted from the one that consists of the measured characteristic maps. Therefore, a good comparison of both is basically given. The results of the comparison are shown in Figure 2.9. Due to the small deviation in vehicle performance between both of the models, the proposed parameterization is considered as accurate enough and used further in this work.

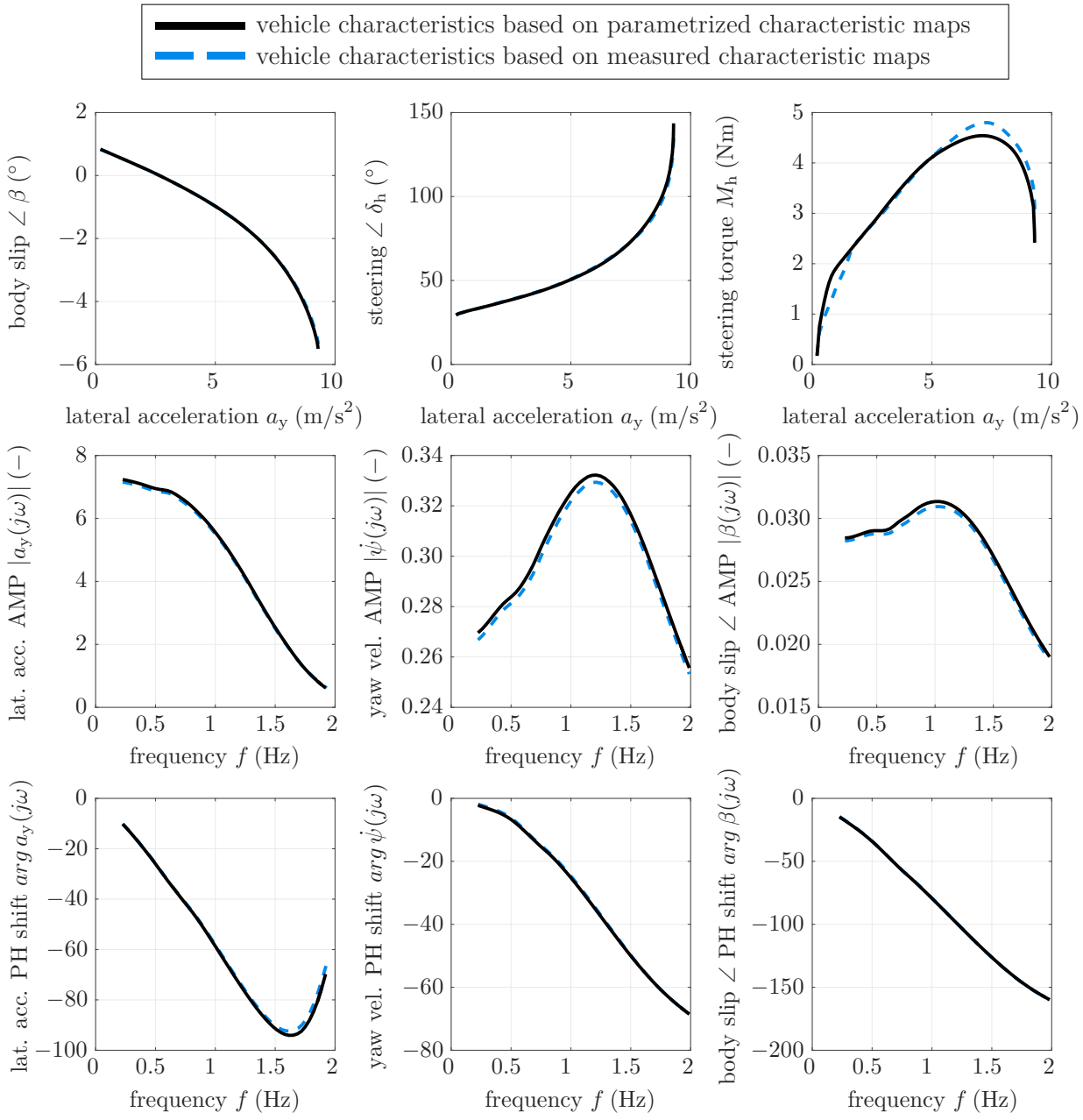


Figure 2.9: Comparison of the system responses between a vehicle model based on measured characteristic maps and those which are based on the functional suspension properties.

Steering The decision for a particular steering system depends on many criteria such as geometrical integration, performance, and cost. From a functional point of view, the maximum available power and force are the two decisive factors in the early development phase. Therefore, Figure 2.10 presents an overview of the ranges of applications for various steering gear types. Since the development and integration of a new steering gear is a long-running project, the decision for a specific type as well as necessary power and force is made in the early development phase. Due to the complex causal relationships between

the steering system, suspension and tires the methods presented in this work offer great potential for consistent design and making correct decisions.

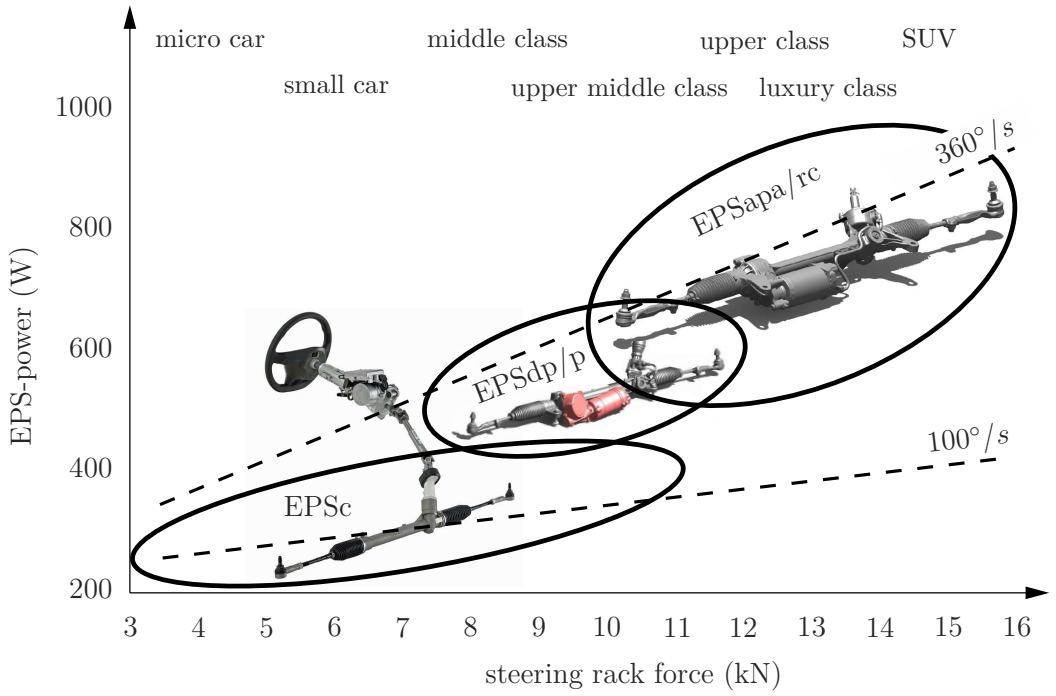


Figure 2.10: Ranges of application for various steering gear types based on (Harrer and Pfeffer 2016). Image of EPSc from (Mareis 2014).

The power curve regarding the steering rack is a relevant property for vehicle system design. In this work, the power characteristics are reduced to three design variables, the maximum power $P_{\mathrm{EPS,\,max}}$, the maximum rack force $F_{\mathrm{rack,max}}$ and the rack speed at which the power decreases v_{dec} as shown in Figure 2.11 (1).

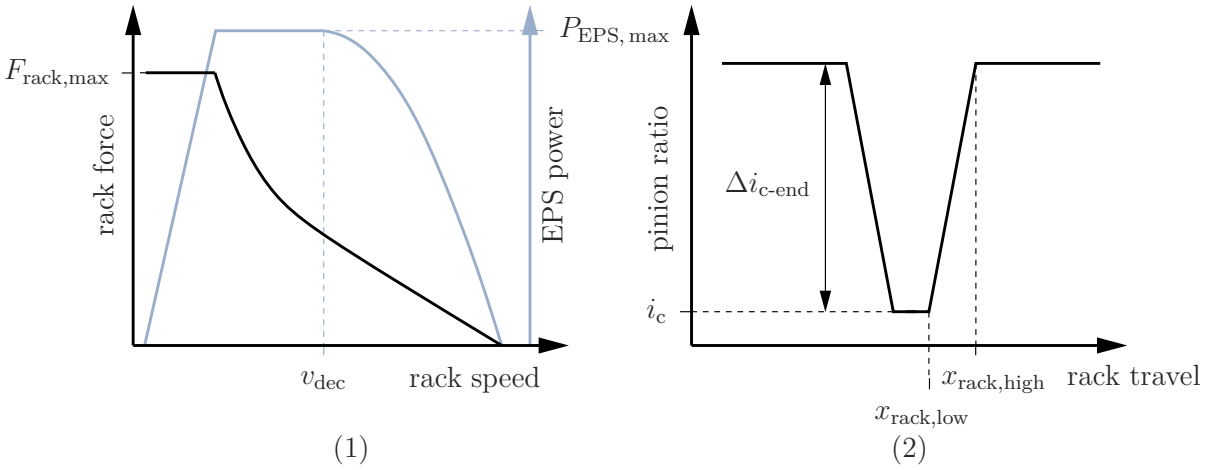


Figure 2.11: Parameterization of the EPS system.

In addition, the available rack force depends on the inertia of the EPS motor and the drive train, therefore, the reduced mass of the steering rack m_{red} is considered as well.

Another important property of the EPS itself, which influences the required steering power, is the pinion ratio, also called pinion to rack ratio. The selected parameterization is shown in Figure 2.11 (2) for a variable ratio.

The design variables belonging to the steering system are comprised in Table 2.3. As mentioned for the suspension design, a detail level also exists for the steering system design. However, the *Original Equipment Manufacturer* (OEM) usually does not develop the steering system on his own, and therefore does not have enough information and detailed simulation models in order to evaluate the feasibility of the requirements.

Table 2.3: Overview of design variables belonging to the steering system.

design variable	**symbol**	**unit**
maximum available steering rack force	$F_{\text{rack,max}}$	N
maximum EPS power	$P_{\text{EPS, max}}$	W
reduced mass of the steering rack	m_{red}	kg
power decrease speed of the EPS	v_{dec}	m/s
pinion ratio at the centered steering rack position	i_{c}	mm/rev
maximum difference of the pinion ratio	$\Delta i_{\text{c-end}}$	mm/rev
steering rack stroke until pinion ratio is equal i_{c}	$x_{\text{rack,low}}$	mm
steering rack stroke from than on pinion ratio is equal $i_{\text{c}} + \Delta i_{\text{c-end}}$	$x_{\text{rack,high}}$	mm
maximum steering rack stroke	$x_{\text{rack, max}}$	mm
friction of the steering rack	$F_{\text{frick,rack}}$	N

Tire Two different tire models are used by the vehicle model depending on the considered objective criteria, one for the evaluation of vehicle dynamics and another one for parking maneuvers.

The tire model for *vehicle dynamics* is the MF-tire model, which was introduced by Pacejka and Bakker (1992). Since that time, it was further developed to take more effects into account such as tire pressure. Up to 134 parameters are available for adjusting the characteristics of the virtual tire to represent the "real world" tire behavior. The parameters of the MF-model need to be adapted such that the difference between real and virtual tire is minimized. Different procedures were previously developed for measuring the characteristics of a tire in a standardized way (TNO 1999). After measuring the tire characteristics, they may be fitted to the virtual model, therefore, numerical optimization may be used for parameter identification (Automotive 2008). Usually, a tire that already exists in hardware is measured first and then the design variables of the virtual tire are adapted such that certain data points of the measured characteristic curves of the tire are equal to those of the virtual tire. The tuning of the MF-parameters according to required FTCs, also called Tire-CVs, is described in (Niedermeier, Peckelsen, and Gauterin 2013). Further details about the MF-tire model are described in (Pacejka 2006).

The tire model for *parking maneuvers* is an in house developed model by BMW. Although it is based on basic simple equations, it is useful in the top-down requirement process. The underlying reason for this is that it is not about describing the behavior of an existing tire based on exact properties such as tire width, diameter, pressure etc., but describes how the tire should behave independently of those properties. Therefore, the tire manufacturer that has its core competence in designing a tire according to certain tire characteristics can focus on designing tires, while the OEM can focus on the system design. The advantage of the applied tire model for parking compared to more complex tire models such as FTire or CDTire is that the parameterizing procedure is less complex, which is helpful especially in the early development stage when limited information is available. Therefore, the model computes the drilling torque based on a reduced set of design variables as shown in Figure 2.12. It estimates the load distribution of the tire and computes the forces and moments equilibrium around the kingpin axis. Combined with the kinematic relations of the suspension, the steering rack force may be calculated.

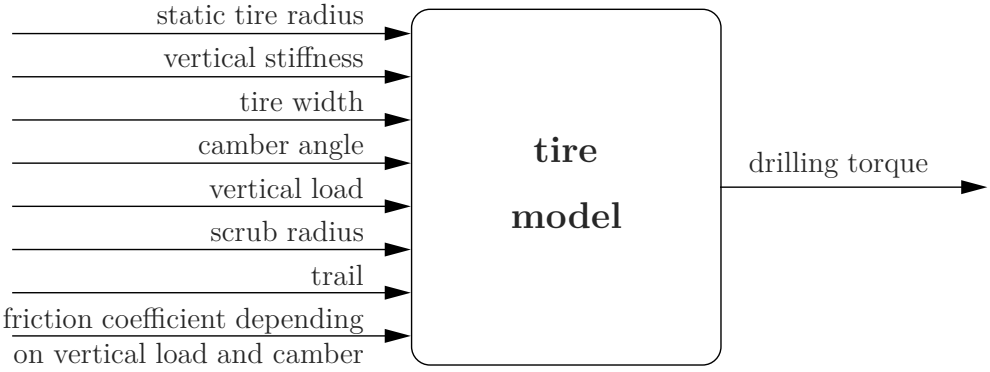


Figure 2.12: Input/output box of the tire model used for parking.

2.2.3 Artificial Vehicle Model

Efficiency plays an important role within the design procedure of a new product. Therefore, information must be available on time to influence the decisions made in the project team. Although the presented physical model is quite fast, it is important to reduce the calculation time even further in order to make it applicable for advanced design approaches that rely on a high number of computations, e.g., a stochastic solution space algorithm. In order to increase the efficiency of the algorithms presented in Chapter 5, surrogate models may be used. Surrogate models help to reduce the computational effort by reducing the time expense of repetitive time-consuming simulation tasks. Nevertheless, creating surrogate models takes some effort as well, which must be taken into account. In this work, an efficient approach for creating fast computing surrogate models by use of machine learning is presented. Therefore, five sequential steps are presented in Figure 2.13 and described in this subsection.

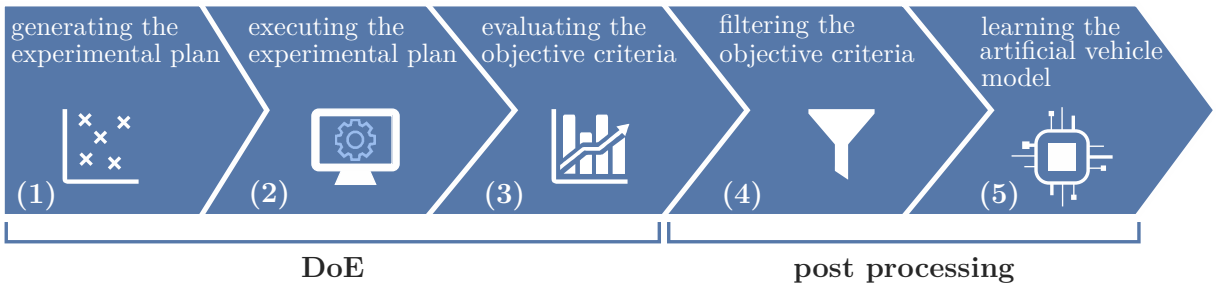


Figure 2.13: DoE based creation of an artificial vehicle model by help of machine learning.

Design of Experiment

DoE describes an experimental plan that may be executed by simulation or even hardware testing. The goal of a DoE is to gather information about the relationships between design variables and objective criteria. Depending on the number of input variables, as well as the non-linearity of the response, more or less sample points are necessary in order to maintain valid information about the system behavior. In order to maintain reliable information about the system behavior, the sample points should be evenly distributed within the design space. The quality of the sample point distribution within a hypercube may be measured by the discrepancy index (Morokoff and Caflisch 1994). In order to increase the quality of the sample point distribution and therefore reduce the number of necessary sample points, several techniques for creating statistical test plans are available, such as Monte Carlo, Latin Hypercube, Full Factorial, Halton-Sequence and Sobol Sequence sampling.

Requirements on the DoE Method. In this work, the applied DoE method needs to satisfy two main requirements: First, the user must be able to select the number of sample points, and second, it must be possible to filter certain sample points following the DoE procedure. The sample points need to be filtered following the DoE procedure, as certain parameter combinations may lead to designs that are not able to perform the maneuver and therefore can not be evaluated. For selected objective criteria the applied filter criteria are presented in Appendix A.3. Since all sample points are necessary for applying Halton Sequence or Sobol Sequence sampling, they are not suited for this work. Full-factorial sampling has the highest evaluation effort and is not suited for a large number of continuous design variables. Therefore, the user can select between Monte Carlo and Latin Hypercube sampling. The difference between both sampling techniques is that Monte Carlo sampling generates evenly distributed random variables with the following variable being independent of the previous one and Latin Hypercube sampling additionally systematically distributes them for better distribution of the samples. Therefore, Latin Hypercube sampling reduces the number of required sample points, however, Monte Carlo sampling is more often used and easier to apply.

Generating an Artificial Vehicle Model based on Machine Learning.

The artificial vehicle model is based on supervised machine learning. In order to create such a model, the following steps are necessary. First, the input parameters must be varied and the simulations performed in order to obtain the performance values. Second, the results must be filtered. Especially when very sensitive parameters are varied over a wide range, discontinuities could result in the system response, e.g., unstable vehicles that spin or overturn. Those results must be filtered to receive an accurate surrogate model. The missing data is not a problem, as the filtered vehicles are far from being good, and the surrogate model will detect them as bad vehicles. In the third step, the surrogate model is created. Therefore, a neural network is trained based on the filtered input/output data. The data is split into the following three groups: training (70%), validation (15%) and testing data (15%). For training, the Levenberg-Marquardt algorithm is used to maximize the absolute value of the coefficient of determination R^2 of the validation data between original and surrogate model. The neural network with the highest R^2 value is used for the design methods described in Chapter 5.

Accuracy of the Artificial Vehicle Model. Evaluating the accuracy of the artificial vehicle model is different compared to the physical vehicle model. The goal of a physical vehicle model is to minimize the deviation between a performance measure that was evaluated during real vehicle tests and virtual vehicle tests under the same circumstances. In contrast, the reference for the accuracy of the artificial vehicle model is the physical vehicle model. The broader the application bandwidth of the artificial vehicle model, the more difficult it is to accurately represent the same behavior as the physical vehicle model. Therefore, the larger the number of design variables to be considered as tunable parameters within the artificial vehicle model and the bigger the considered design space, the larger the number of necessary sample points to learn the vehicle behavior.

2.3 Design Algorithms for Vehicle Architecture Design

Due to an ongoing shift of research fields in the automobile industry, from hardware based to software based, such as artificial intelligence, digitalization or cloud services, the vehicle manufactures are forced to increase their efficiency within the classical design fields without neglecting quality. Accordingly, high demands are set on contemporary development processes that also need to be adapted to digitalization. In an uncertain environment, flexibility of a design during the development process as well as robustness toward uncertainties is the key to success. Therefore, both must be considered by a contemporary design process. If the robustness of a favored design is low, the flexibility must be high in order to compensate and vice versa.

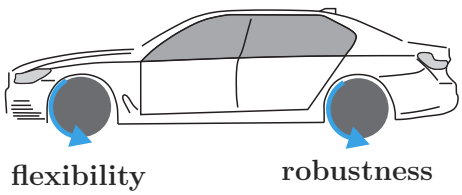


Figure 2.14: Importance of flexibility and robustness within the design process.

This can be compared to a four-wheel drive vehicle where the stronger driven axle must compensate the weaker one, such that the vehicle is able to achieve a certain longitudinal acceleration (see Figure 2.14). Nevertheless, it would be preferable if both axles would be strong driven.

Definition 3 (flexibility)
In the context of a development process, flexibility *denotes the capacity to adapt a favored design due to changed requirements or a changed environment with little effort.*

The effort is measured relative to the current phase of the product in the development process. The earlier in the development phase, the more flexibility is required in order to find a consistent design that satisfies the needs of the customer. Therefore, back-tracking in the development process becomes less expensive.

Flexibility may be ensured by shorter iteration loops and the consideration of multiple designs during the development process. Shorter iteration loops may be accomplished by an increased use of CAE and could reduce the expenses of back-tracking during the development process. Considering multiple designs during the development process enables the project team to select between different alternative options at each stage of the development process without time delay.

Definition 4 (robustness)
When considering a particular good design within the design space $\boldsymbol{y}_{\text{good}} \in \Omega_{\text{ds}}$*, the robustness is the allowed deviation of the design variables such that the required performance is still satisfied.*

According to Definition 4, robustness is measured in the input space. A robust design is able to meet the customer expectations, even if certain deviations from the intended design occur. Therefore, back-tracking in the development process is reduced as well. Robustness can be increased by considering uncertainties during the development process. Uncertainties might be manufacturing margins or changing environments as well as requirements. However, the consequences of the various uncertainties are the same, as they all complicate meeting the customers expectations. As described in Section 1.2, certain design methods, such as robust design optimization and solution spaces, take uncertainties into account and reduce the risk of failing the customer expectations.

In order to guarantee both flexibility and robustness, solution spaces and multidisciplinary multidimensional robust design optimization are combined in this work.

2.3.1 General Design Strategy for the Vehicle Architecture

As already mentioned in Section 1.5, the vehicle is clustered hierarchically into three abstraction levels. The different abstraction levels distinguish between requirements on the system, subsystem, and detail properties of the overall system. For reasons of traceability, feasibility restrictions of subsystem properties occurring from the causal relationships between properties of the subsystem and detail level will not be treated yet. The aim of the

Vehicle Architecture Design is to derive requirements top-down from the system level to the subsystem level. Therefore, a simulation model, e.g., as described in Section 2.2, which computes the system performance as a function of the subsystem properties $\boldsymbol{z} = \mathrm{f}_1(\boldsymbol{y})$, may be used in combination with an appropriate optimization technique in order to invert the computation such that $\boldsymbol{y} = \mathrm{f}_1{}^{-1}(\boldsymbol{z})$. Resulting from the requirements regarding robustness and flexibility on the design sought, which were described at the beginning of this section, solution spaces will be identified. Therefore, an alternative novel method for identifying box-shaped solution spaces will be applied.

The properties at the system level $\boldsymbol{z}$ are called *characteristic values* (CVs). For those objective criteria, permissible intervals $I_i^{\mathrm{z}} = [z_i^{\mathrm{lb}}, z_i^{\mathrm{ub}}]$ are prescribed by an engineer based on the expectations of the customer, where i is the counting variable of the objective criteria from 1 to the number of objectives o. The target area of the system performance $\Omega_{\mathrm{sb}}^{\mathrm{z}}$ is defined by the Cartesian product of the target intervals, see Equation (2.2).

$$\Omega_{\mathrm{sb}}^{\mathrm{z}} = I_1^{\mathrm{z}} \times I_2^{\mathrm{z}} \times \cdots \times I_o^{\mathrm{z}} \tag{2.2}$$

As shown in Figure 2.15, the shape of the total solution space $\Omega_{\mathrm{ss}}^{\mathrm{y}}$ depends on the prescribed target area $\Omega_{\mathrm{sb}}^{\mathrm{z}}$ as well as the causal relationships $\mathrm{f}_1(\boldsymbol{y})$ between subsystem and system properties. Since describing the total solution space may become very complicated in higher dimensions and also has several disadvantages, such as visualization problems, computational cost and coupled design variables, only a box shaped solution space (solution box) is identified.

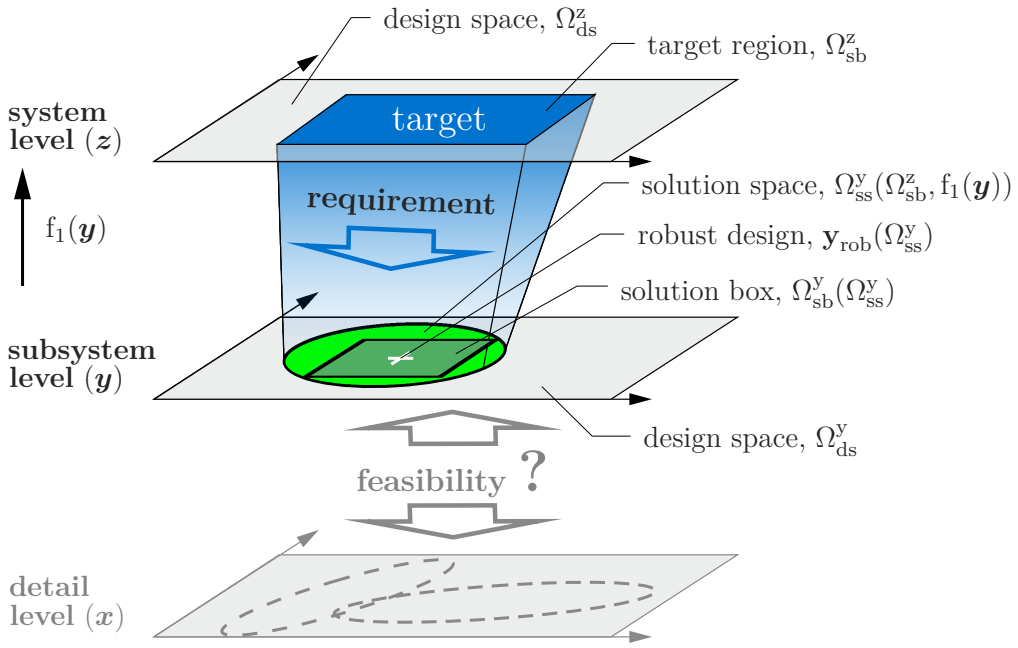


Figure 2.15: General design strategy for the Vehicle Architecture.

For identifying the biggest solution box regarding certain constraints, two sequential numerical optimizations are performed only once. The first optimization identifies a robust design $\boldsymbol{y}_{\mathrm{rob}}^{y}$ based on the target area of the system level properties by approximating the

total solution space. The method is further described in Subsection 2.3.3. Therefore, the computational burden resulting from the migration of the solution box may be further reduced, since the starting area for identifying the solution box is better suited. In the following, the precise shape of the biggest solution box is identified by numerical optimization starting from the previously determined robust design as described in Subsection 2.3.4. Within the calculated permissible intervals on the subsystem level, all top level requirements are satisfied. This leads to flexibility within those intervals and robustness against variations, if a particular design should be realized.

2.3.2 Concept of Solution Spaces

The mindset in the design procedure using solution spaces is fundamentally different from classical point based design. In the design procedure using solution spaces, the design space is divided into two groups: one group includes all the good designs which satisfy all requirements on the product, the other group includes all bad designs which fail at least one requirement. A particular design is evaluated by threshold values regarding its system responses. The threshold values can either be lower bounds z^{lb}, upper bounds z^{ub} or both $z^{\mathrm{lb/ub}}$ and are expressed in cumulated form by f_c. All threshold values together form the target area $\Omega^{\mathrm{z}}_{\mathrm{sb}}$. If the performance of a particular design lies within the target area, it is called a good design.

Definition 5 (target area, $\Omega^{\mathrm{z}}_{\mathrm{sb}}$)
The target area is a subspace in the output space which defines the required performance of good vehicles.

In a top-down process, the overall performance requirements are described by the target region. Although arbitrary volumes might be possible as target region, a hyperbox is usually used. Therefore, they are similar to solution boxes in comparison with the lower level. The area within the design space $\Omega^{\mathrm{y}}_{\mathrm{ds}}$ that includes all good designs is the so called solution space $\Omega^{\mathrm{y}}_{\mathrm{ss}}$.

Definition 6 (design space, $\Omega^{\mathrm{y}}_{\mathrm{ds}}$)
The design space defines for all design variables the possible values they are allowed to accept, and therefore comprises all possible designs as well as the maximum borderlines of the solution space. The design space should not be selected too small in order to avoid excluding alternative solutions.

Definition 7 (solution space, $\Omega^{\mathrm{y}}_{\mathrm{ss}}$)
The solution space comprises all designs within the design space that satisfy all requirements imposed from the level above.

Due to the non-linear and interlinked nature of the causal relationships between design variables and objective criteria, the solution space usually has an arbitrary shape which is not easy to describe. Primarily design problems with a large number of objective criteria

and requirements easily become complex and difficult to evaluate. By reducing the information on whether a design is good or bad, the focus is laid on the important message, which results in improved clarity.

Statement 2 (clarity)
Within the solution space, all possible combinations of design parameters deliver a good design in terms of satisfying all specified requirements. Since designs are only distinguished between being good or bad, all designs within the solution space are treated equally.

A solution box $\Omega_{\mathrm{sb}}^{\mathrm{y}}$ is laid within the solution space such that all top-level requirements are satisfied including $\Omega_{\mathrm{sb}}^{\mathrm{y}} \subseteq \Omega_{\mathrm{ss}}^{\mathrm{y}}$.

Definition 8 (solution box, $\Omega_{\mathrm{sb}}^{\mathrm{y}}$)
A solution box is a box shaped solution space and therefore a subset of the solution space.

Within the solution space, infinite solution boxes of different size and position might be laid. However, to quantify how good a particular solution box is suited, the box size measure μ is used. In general, the volume of the box is used to measure its size (Zimmermann and Hoessle 2013; Fender et al. 2014; Eichstetter, Müller, and Zimmermann 2015)) which is calculated by the Cartesian product of the permissible intervals, see Equation (2.3).

$$\mu\left(\Omega_{\mathrm{sb}}^{\mathrm{y}}\right) = \mu\left(I_1^{\mathrm{y}} \times I_2^{\mathrm{y}} \times \cdots \times I_d^{\mathrm{y}}\right) = \prod_{j=1}^{d} \left(y_j^{\mathrm{ub}} - y_j^{\mathrm{lb}}\right) \tag{2.3}$$

Therefore, y denotes the corresponding abstraction level on which the box size measure is computed and j is the counting variable from 1 to the number of dimensions d.

Using solution boxes offers advantages, but also disadvantages which are summarized in Table 2.4.

Thus far, the main goal of the design using solution spaces is finding the solution box with the maximum size measure under consideration of different constraints. Therefore, numerical optimization may be used. The general problem statement for finding such a solution box on the subsystem level reads as follows: Adapt the intervals of the solution box $\Omega_{\mathrm{sb}}^{\mathrm{y}}$ such that the box size measure μ is maximized, while the values of all objective criteria of all possible designs within $\Omega_{\mathrm{sb}}^{\mathrm{y}}$ are below a critical threshold value, see Equation (2.4).

$$\underset{\Omega_{\mathrm{sb}}^{\mathrm{y}}}{\text{maximize}}\ \mu\left(\Omega_{\mathrm{sb}}^{\mathrm{y}}\right)\ \text{s.t.}\ \mathrm{f}_1\left(\boldsymbol{y}\right) \leq \mathrm{f}_{1,\mathrm{c}}\ \forall\, \boldsymbol{y} \in \Omega_{\mathrm{sb}}^{\mathrm{y}} \subseteq \Omega_{\mathrm{ds}} \tag{2.4}$$

Various possibilities exist to attenuate the drawbacks of using box shaped solution spaces, to name two of them:

- The solution box may be rotated by a rotation matrix in order to increase the volume of the solution box. Therefore, the rotation angle needs to be integrated as an additional optimization parameter. Daub and Duddeck (2018) apply rotated ellipses to exploit a greater volume of the solution space.

- Erschen (2017) applies 2-dimensional maps to combine particular design variables in order to increase the volume of the exploited solution space.

Table 2.4: Advantages and disadvantages of laying a solution box in the solution space.

+	−
• A solution box can be described easily by intervals. • A solution box may be computed for high dimensions with affordable effort and high reliability. • Every combination of parameters within the solution box satisfies the overall requirements, therefore parameters are decoupled, which supports the cooperation of multiple departments. As long as the performance of a particular component designed by department A lies within the associated permissible intervals I, they do not need to communicate with department B.	• If the solution space has the shape of a thin strip, solution boxes only consider a small fraction of all possible solutions. Within high dimensions, this problem becomes greater. • The solution space outside of the box is not considered for the detail design on the x-level as long as the solution spaces are not shifted with respect to feasibility restrictions.

Both of the mentioned methods create the same problem, which is that design variables get coupled and cannot be adjusted independently of each other within a permissible interval. Therefore, decentralized development becomes more difficult such that "classical" (not rotated) solution boxes should be used, if they provide enough robustness and flexibility, which is the case in this work. However, depending on the design problem, coupling of design variables may be a useful tool in order to increase the exploited solution space.

Simple Example of a Solution Space in Vehicle Design

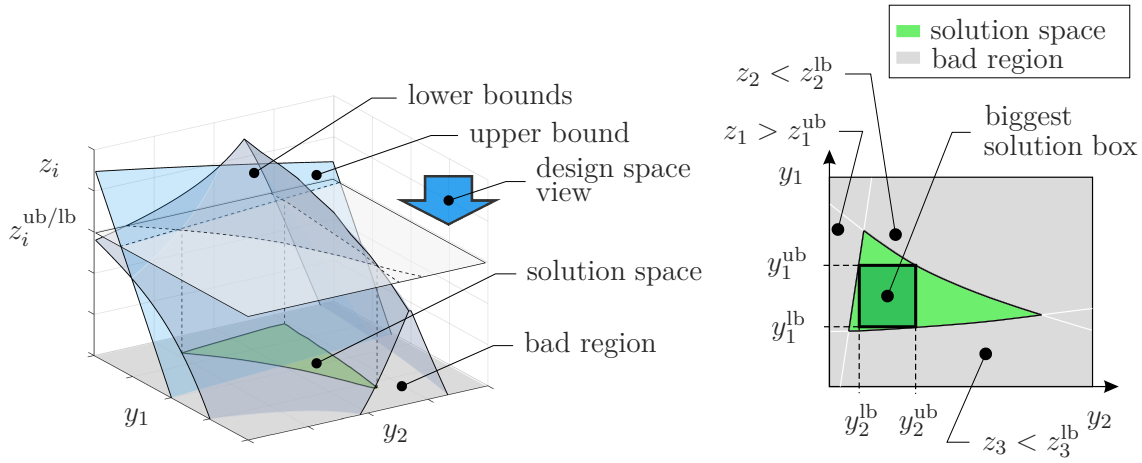
In order to explain the basic concept in a practical way, a simple example out of the field of vehicle dynamics is presented in Figure 2.16 (a). The example considers three different system responses z_1, z_2 and z_3 as well as two design variables y_1 and y_2, which are explained in Table 2.5. For each system response, a particular threshold value is defined, which leads to a bad design, if the actual response lies above or beyond the threshold. Therefore, the system response z_1 leads to a bad design if the threshold value z_1^{ub} is exceeded, while the responses of z_2 and z_3 lead to a bad design if their value does not reach the associated threshold value z_2^{lb} and z_3^{lb}. Based on the three threshold values z_1^{ub}, z_2^{lb} and z_3^{lb}, the design

space may be divided into good and bad designs, such that the two dimensional solution space shown in Figure 2.16 (b) is identified.

Table 2.5: Physical parameters of the generalized system responses $\boldsymbol{z}$ and the design variables $\boldsymbol{y}$.

Variable	Symbol	Physical parameter
z_1	$T_{a_y-\dot{\psi}}$	time delay between lateral acceleration and yaw velocity response
z_2	$F_{\mathrm{z,\,min}}$	minimum wheel load while steady-state cornering
z_3	$a_{\mathrm{y,\,max}}$	maximum lateral acceleration
y_1	$\mu_{\mathrm{y,\,max}}$	lateral friction coefficient on all four tires
y_2	$TLLTD$	total lateral load transfer distribution

For computing the size of the solution box, the two intervals $I_1 = [y_1^{\mathrm{lb}}, y_1^{\mathrm{ub}}]$ and $I_2 = [y_2^{\mathrm{lb}}, y_2^{\mathrm{ub}}]$ need to be multiplied by each other. The solution box with the biggest box size measure is indicated in Figure 2.16b as well. It is also possible to compute solution boxes for problems with more than two design parameters, however their verification, visualization and comparison with alternative solution boxes is a challenge during the design procedure. Thus, if all those challenges can be handled, solution boxes offer great potential for the decentralized development of subsystems. However, the quality, significance and accuracy of the result delivered by this method strongly depend on the quality of the input data as well as simulation results. In addition, it is of great importance that the requirements on the top-level objective criteria correspond to the subjective evaluation of the vehicle in order to meet the customer expectations.



(a) response of multiple objectives on a variation of two design variables

(b) two dimensional solution space in the design space view

Figure 2.16: Principal of solution spaces.

2.3.3 Robust Optimization

A *Robust Optimization* may be performed in order to identify an appropriate starting point for the following solution box optimization. Therefore, a design point is sought such that the distance to the closest performance threshold value, which leads to design failure, is maximized within the design space. To measure the distance between a particular design point and the performance borderlines within the design space, linear approximation of the underlying model is used. The linear approximation results in low computational costs, however, approximations can lead to inaccuracies and therefore incorrect results. Nevertheless, those inaccuracies are acceptable for the following reasons:

- The design point resulting from the robust optimization is used as the initial design point for the solution box optimization which then computes the borderlines of the solution box more accurately. Therefore, based on the robust optimization, no absolute statement concerning the allowed deviation from the design point is made.
- Although non-linearities between design variables and objective criteria are quite reasonable in the field of vehicle dynamics, they are not reasonable between different design variables.
- Errors caused by inaccuracies resulting from non-linearities between different design variables depend on the distance to the linearized state. Therefore, the safety margin of a design with low robustness may be calculated more accurately than those of a design with high robustness.

How to Measure the Robustness of a Particular Design Point?

In order to combine robust design optimization with solution spaces by a sequential optimization, certain requirements regarding the efficiency and objective criteria of the method exist. Therefore, the way of approximating the robustness of a particular design point will be described first.

Definition 9 (robust design, $\boldsymbol{y}_{\mathrm{rob}}$)
A robust design is a particular design point within the solution space that satisfies the performance requirements, although environmental boundary conditions may differ (noise, temperature, manufacturing, etc.). The robustness of a design is measured by the distance to the borderlines of the solution space. The distance is measured by μ_{rb}*, see* Equation (2.15).

To describe the method, a simple example considering one design variable and two objective criteria is used, see Figure 2.17. Therefore, the actual solution space is illustrated in Figure 2.17 (a). As the method may be applied to design problems with multiple design variables in the same way, y_j is used as a general design variable where j is the counting variable from 1 to the number of dimensions d.

Both objective criteria include two threshold values each, an upper bound z_i^{ub}, and a lower bound z_i^{lb}. Therefore, the performance of both z_1 and z_2 needs to be within a specified range to maintain a good design. The design variable can be shifted into two directions until the design fails a first objective. Therefore, according to Figure 2.17 (a) the actual one-dimensional solution space may be described by the interval $I_j = [y_j^{\mathrm{lb},z_1}, y_j^{\mathrm{ub},z_2}]$.

However, the fundamental difference between both objective criteria is that z_1 is negatively correlated to the design variable y_j, while z_2 is positively correlated to it. Therefore, the relations between the design variables $\boldsymbol{y}$ and objective criteria $\boldsymbol{z}$ can be clustered into the following four categories.

- Relations between upper bounds of objective criteria and positively correlated design parameters (Figure 2.17 (a): z_2^{ub} and y_j^{ub,z_2}).
- Relations between lower bounds of objective criteria and negatively correlated design parameters (Figure 2.17 (a): z_1^{lb} and y_j^{ub,z_1}).
- Relations between upper bounds of objective criteria and negatively correlated design parameters (Figure 2.17 (a): z_1^{ub} and y_j^{lb,z_1}).
- Relations between lower bounds of objective criteria and positively correlated with parameters (Figure 2.17 (a): z_2^{lb} and y_j^{ub,z_2}).

For computing the robustness margin, one must consider that lower bounds of design variables result from lower bounds of positive correlated objective criteria and upper bounds of negative correlated objective criteria. Therefore, upper bounds of design variables result from upper bounds of positive correlated objective criteria and lower bounds of negative correlated objective criteria.

A design is called *good* if its performance lies within the target range of both objective criteria such that all requirements on the design are satisfied. Therefore, two threshold values within the design space are of great importance, the one that leads to design failure when the design variable is increased and the one that leads to failure when the design variable is decreased. Those critical threshold values within the input space are called $y_{\mathrm{rob},j}^{\mathrm{lb}}$ and $y_{\mathrm{rob},j}^{\mathrm{ub}}$.

In the case of Figure 2.17 (a), $y_{\mathrm{rob},j}^{\mathrm{lb}}$ is equal y_j^{lb,z_1} and $y_{\mathrm{rob},j}^{\mathrm{ub}}$ is equal y_j^{ub,z_2} as they are the closest threshold values within the design space. In order to increase the robustness of a design, the distance to the borderlines of the solution space (also called robustness intervals) needs to be maximized, such that

$$|\Delta y_{\mathrm{rob},j}^{\mathrm{lb}}| \wedge |\Delta y_{\mathrm{rob},j}^{\mathrm{ub}}| \rightarrow \max . \tag{2.5}$$

Computation of the robustness intervals. As presented in Figure 2.17 (b), at first, the threshold values of the objective criteria $\boldsymbol{z}^{\mathrm{lb}} = [z_1^{\mathrm{lb}}, z_2^{\mathrm{lb}}]$ and $\boldsymbol{z}^{\mathrm{ub}} = [z_1^{\mathrm{ub}}, z_2^{\mathrm{ub}}]$ are neglected such that it is assumed that all possible values of y_j within the design space result in a good design. The reason for this is that if a particular direction of the solution space is not restricted by any objective, it needs to be restricted by the design space. Therefore, the distances of $\Delta \boldsymbol{y}_{\mathrm{rob}}$ are restricted such that $\boldsymbol{y}_{\mathrm{rob}}^{\mathrm{lb}}$ and $\boldsymbol{y}_{\mathrm{rob}}^{\mathrm{ub}}$ lie in the design space.

A particular direction of a design variable may need to be restricted by the design space for two reasons: First, no objective exists restricting it, and second, the design variable has a very low sensitivity to all objective criteria that would result in a restriction in this direction. In the second case, the critical threshold value of the particular design variable resulting from the associated design objectives would lie outside of the design space.

Therefore, the distances between the current design $\boldsymbol{y}$ and the associated design space limits $\boldsymbol{y}^{\mathrm{ds,lb}}, \boldsymbol{y}^{\mathrm{ds,ub}}$ are calculated as shown in Equations (2.6) and (2.7) by subtracting them.

$$\Delta\boldsymbol{y}^{\mathrm{ds,lb}} = \boldsymbol{y} - \boldsymbol{y}^{\mathrm{ds,lb}} \tag{2.6}$$

$$\Delta\boldsymbol{y}^{\mathrm{ds,ub}} = \boldsymbol{y} - \boldsymbol{y}^{\mathrm{ds,ub}} \tag{2.7}$$

In the following, the threshold values resulting from the objective criteria are considered as well. For identifying the distance between the current design point $\boldsymbol{y}$ and the threshold values of the objective criteria within the design space, linear approximation may be used. The linearizion is applied at the design point $y*_j$ of the current iteration loop in order to ensure a good accuracy in the local environment, see Figure 2.17 (c). Therefore, for each design variable the deviation of all performance measures is computed based on a variation of 1% of the associated design interval $I_j^{\mathrm{ds}} = [y_j^{\mathrm{ds,lb}}, y_j^{\mathrm{ds,ub}}]$. The resulting gradients of all the input/output relations are comprised in the $n_{\mathrm{inputs}} \times n_{\mathrm{outputs}}$-sensitivity-matrix $\boldsymbol{A}$. Therefore $\boldsymbol{a}_j$ is a particular row of the $\boldsymbol{A}$-matrix including the sensitivities of the j-th design variable regarding all objective criteria.

Based on the Equations (2.8) and (2.9), the distance between the j-th design variable and each threshold value is calculated within the design space.[5] Therefore, for one particular design variable, the vector $\Delta\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{lb}}$ comprises all robustness intervals resulting from lower bounds of objective criteria, while $\Delta\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{ub}}$ comprises all robustness intervals resulting from upper bounds of objective criteria.

$$\Delta\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{lb}} = \frac{\boldsymbol{z}^{\mathrm{lb}} - \boldsymbol{z}}{\boldsymbol{a}_j} \tag{2.8}$$

$$\Delta\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{ub}} = \frac{\boldsymbol{z}^{\mathrm{ub}} - \boldsymbol{z}}{\boldsymbol{a}_j} \tag{2.9}$$

However, $\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{lb}}$ as $\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{ub}}$ alone do not indicate whether the design variable is restricted from the top or from the bottom. Therefore, the correlation between design and objective criterion must be considered as well. Based on a logical $n_{\mathrm{inputs}} \times n_{\mathrm{outputs}}$-correlation-matrix that distinguishes between positive and negative correlation, it may be distinguished from which side the design variable is restricted. Following the appropriate assignment according to the Equations (2.10) and (2.11), where + indicates a positive correlation and − a negative correlation, $\Delta\boldsymbol{y}_j^{\mathrm{lb}}$ includes all restrictions to the design variable from the bottom and $\Delta\boldsymbol{y}_j^{\mathrm{ub}}$ all restrictions from the top.

$$\Delta\boldsymbol{y}_j^{\mathrm{lb}} = (\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{lb}}(+), \boldsymbol{y}_j^{\boldsymbol{z},\mathrm{ub}}(-)) \tag{2.10}$$

[5] The division is defined element-wise.

$$\Delta \boldsymbol{y}_j^{\mathrm{ub}} = (\boldsymbol{y}_j^{\boldsymbol{z},\mathrm{lb}}(-), \boldsymbol{y}_j^{\boldsymbol{z},\mathrm{ub}}(+)) \tag{2.11}$$

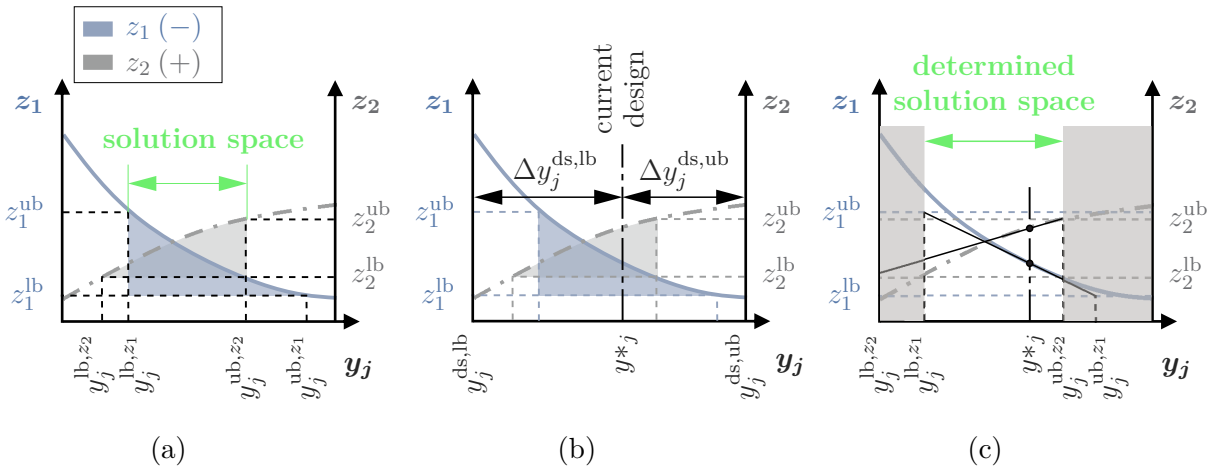


Figure 2.17: One-dimensional example of the proposed method for robust design.

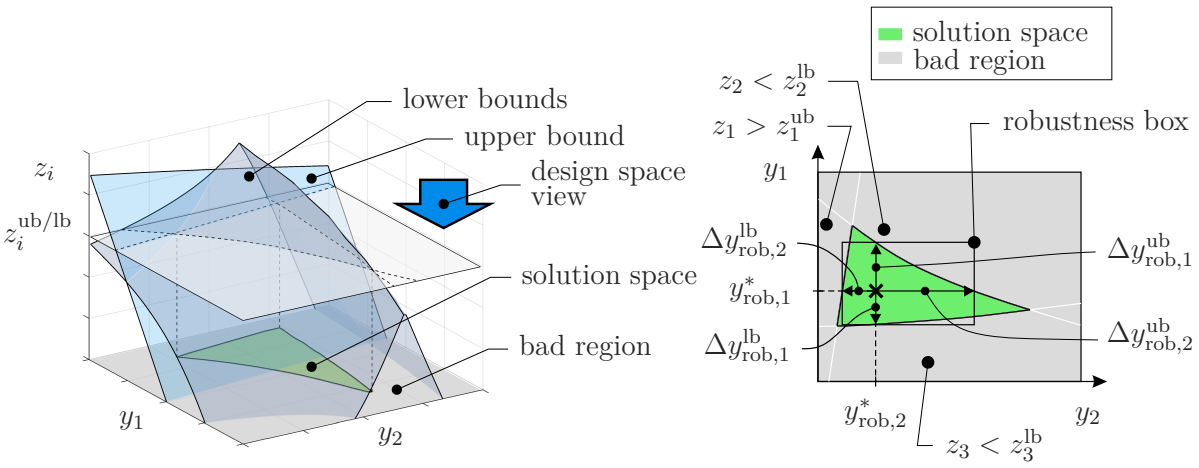
In order to identify the closest/critical threshold values, Equations (2.12) and (2.13) are used. Since restrictions from the bottom have a negative sign, the one with the maximum value is sought, while for restrictions from the top the minimum value is sought.

$$\Delta y_{\mathrm{rob},j}^{\mathrm{lb}} = \max\left(\Delta \boldsymbol{y}_j^{\mathrm{lb}}\right) \tag{2.12}$$

$$\Delta y_{\mathrm{rob},j}^{\mathrm{ub}} = \min\left(\Delta \boldsymbol{y}_j^{\mathrm{ub}}\right) \tag{2.13}$$

Therefore, the robustness interval may be defined as $\Delta \boldsymbol{y}_{\mathrm{rob},j} = \left[\Delta y_{\mathrm{rob},j}^{\mathrm{lb}}, \Delta y_{\mathrm{rob},j}^{\mathrm{ub}}\right]$. In addition, the absolute robustness threshold values may be calculated by $y_{\mathrm{rob},j}^{\mathrm{lb}} = y_{\mathrm{rob},j} + \Delta y_{\mathrm{rob},j}^{\mathrm{lb}}$ for the lower bound and by $y_{\mathrm{rob},j}^{\mathrm{ub}} = y_{\mathrm{rob},j} + \Delta y_{\mathrm{rob},j}^{\mathrm{ub}}$ for the upper bound.

Robustness in multiple dimensions. For design problems with multiple dimensions, the complete solution space may not be described by solution intervals, see Figure 2.18.



(a) response of multiple objectives on a variation of two design variables

(b) two dimensional robust design in the design space view

Figure 2.18: The principal of a robustness box.

Therefore, the absolute robustness threshold values $\boldsymbol{y}_{\text{rob}}^{\text{lb}}$ and $\boldsymbol{y}_{\text{rob}}^{\text{ub}}$ do not represent the borderlines of the solution space. However, they indicate the maximum allowed interval of variation of a particular design variable if all other design variables correspond precisely to their nominal value.

Problem Statement for Robust Design

The goal of the optimization itself is to maximize the size measure μ_{rb} of the robustness box $\Omega_{\text{rb}}(\boldsymbol{y})$ by adapting the design variables y_j with $j = 1,2,...,d$. The robustness box Ω_{rb} is a sub-space of the design space Ω_{ds}. Contrary to the solution box Ω_{sb}, not all designs within Ω_{rb} satisfy the requirement $\boldsymbol{z}^{\text{lb}} \leq \text{f}_1(\boldsymbol{y}) \leq \boldsymbol{z}^{\text{ub}}$.

However, the robustness box maximizes μ_{rb} under the constraint $\boldsymbol{z}^{\text{lb}} \leq \text{f}_1(\boldsymbol{y}) \leq \boldsymbol{z}^{\text{ub}}$, assuming that only one design variable y_i is allowed to vary at one time, which also reduces the computational effort compared to solution spaces.

The performance function for the vehicle/robustness box is presented in Equation 2.15. To force the resulting robust design $\boldsymbol{y}_{\text{rob}}$ to lie close to the center of the robustness box Ω_{rb}, the normalized[6] volume is not used as size measure of the box as it is for the solution box optimization. Instead, according to Equation (2.14), the size measure μ_{rb} is computed by the product of the weighted and normalized intervals between the robust design, and the borderlines of the robustness box, $\Delta y_{\text{rob},j}^{\text{lb}}$ and $\Delta y_{\text{rob},j}^{\text{ub}}$, are used.

$$\mu_{\text{rb}} = \prod_{j=1}^{d} \frac{\Delta y_{\text{rob},j}^{\text{lb}} \, \Delta y_{\text{rob},j}^{\text{ub}}}{y_{\text{ds},j}^{\text{ub}} - y_{\text{ds},j}^{\text{lb}}} \, w_j \tag{2.14}$$

Therefore, the possibilities for alternative designs around $\boldsymbol{y}_{\text{rob}}$ are increased. In addition, the weighting factor w_j provides the possibility to distinguish between the importance of certain robustness intervals. Therefore, a bigger value of w_j automatically results in a broader interval width. However, adjusting w_j is not required for the convergence of the method. If a particular design is not satisfying the requirements derived from the system level, which is the case if $\max\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{lb}}\right) > 0$ or $\min\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{ub}}\right) < 0$, the sum of the normalized distances is used to find a single solution satisfying the requirements. Therefore, designs with a condensed system performance $-\mu_{\text{rb}} \leq 0$ are classified as good designs, and those with $-\mu_{\text{rb}} > 0$ are classified as bad designs.

$$\begin{aligned} &\underset{y_1, y_2, \ldots y_d}{\text{minimize}} \quad -\mu_{\text{rb}}(\Omega_{\text{rb}}) \\ &\text{with} \qquad -\mu_{\text{rb}} = \begin{cases} -\prod_{j=1}^{d} \frac{\Delta y_{\text{rob},j}^{\text{lb}} \, \Delta y_{\text{rob},j}^{\text{ub}}}{y_{\text{ds},j}^{\text{ub}} - y_{\text{ds},j}^{\text{lb}}} \, w_j, & \text{if } \max\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{lb}}\right) \leq 0 \\ & \text{or } \min\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{ub}}\right) \geq 0 \\ \sum_{j=1}^{d} \frac{\left|\Delta y_{\text{rob},j}^{\text{lb}}\right| + \left|\Delta y_{\text{rob},j}^{\text{ub}}\right|}{y_{\text{ds},j}^{\text{ub}} - y_{\text{ds},j}^{\text{lb}}} \, w_j & \text{if } \max\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{lb}}\right) > 0 \\ & \text{or } \min\left(\Delta \boldsymbol{y}_{\text{rob}}^{\text{ub}}\right) < 0 \end{cases} \end{aligned} \tag{2.15}$$

[6] The volume is normalized with respect to the design space measure.

2.3.4 Solution Box Optimization

As described in Subsection 1.2.2, the largest solution box may be sought using different approaches. The goals of the solution box optimization presented in this work are different from the ones found in literature. Since the initial design of the solution box optimization is the result of a robust design optimization, there is no need for a big migration of the solution box. Therefore, numerical multiobjective optimization is used to expand the solution box in order to ensure good accuracy for high dimensional problems. The DoF are the boundaries of the solution intervals y_j^{lb} and y_j^{ub} with $j = 1,2,...,d$, where d denotes the number of dimensions. On the basis of the preceding robust design optimization, it may be assumed that lower bounds of the solution box only vary between the value of the robust design and the lower bound of the design space, while the upper bound of the solution box varies between the robust design and the upper bound, see Equations (2.16) and (2.17).

$$\boldsymbol{I}_{\mathrm{ds}}^{\mathrm{lb}} = \left[\boldsymbol{y}_{\mathrm{ds}}^{\mathrm{lb}}, \boldsymbol{y}_{\mathrm{rob}}\right] \tag{2.16}$$

$$\boldsymbol{I}_{\mathrm{ds}}^{\mathrm{ub}} = \left[\boldsymbol{y}_{\mathrm{rob}}, \boldsymbol{y}_{\mathrm{ds}}^{\mathrm{ub}}\right] \tag{2.17}$$

Therefore, the design intervals are narrowed, and thus computational effort is reduced since less iterations are necessary. However, the biggest solution box still is identified. While lower bounds $\boldsymbol{y}_j^{\mathrm{lb}}$ are allowed to lie within the intervals of $\boldsymbol{I}_{\mathrm{ds}}^{\mathrm{lb}}$, upper bounds should lie within the intervals of $\boldsymbol{I}_{\mathrm{ds}}^{\mathrm{ub}}$. The size of the solution box is measured by the normalized product of solution intervals, see Equation (2.18).

$$\mu_{\mathrm{sb}} = \prod_{j=1}^{d} \frac{y_j^{\mathrm{ub}} - y_j^{\mathrm{lb}}}{y_{\mathrm{ds},j}^{\mathrm{ub}} - y_{\mathrm{ds},j}^{\mathrm{lb}}} \tag{2.18}$$

In general, the bigger the size measure of the solution box, the greater the flexibility and robustness. However, the size of the solution box needs to be restricted, as only good designs are desired to be within the solution box. In order to estimate the fraction of good designs within a solution box, Monte Carlo sampling (Binder 2006) is used, as proposed in (Lehar and Zimmermann 2012). Therefore, a candidate solution box is sampled and evaluated in each iteration. The ratio between bad designs B and the total number of designs N is $B_{\%}$. If the ratio of bad designs exceeds the critical threshold value of accepted bad designs, $B_{\%,\mathrm{crit}}$, an additional punishing term, is added to the performance function. The overall performance of a vehicle/solution box, therefore, is quantified by Equation (2.19).

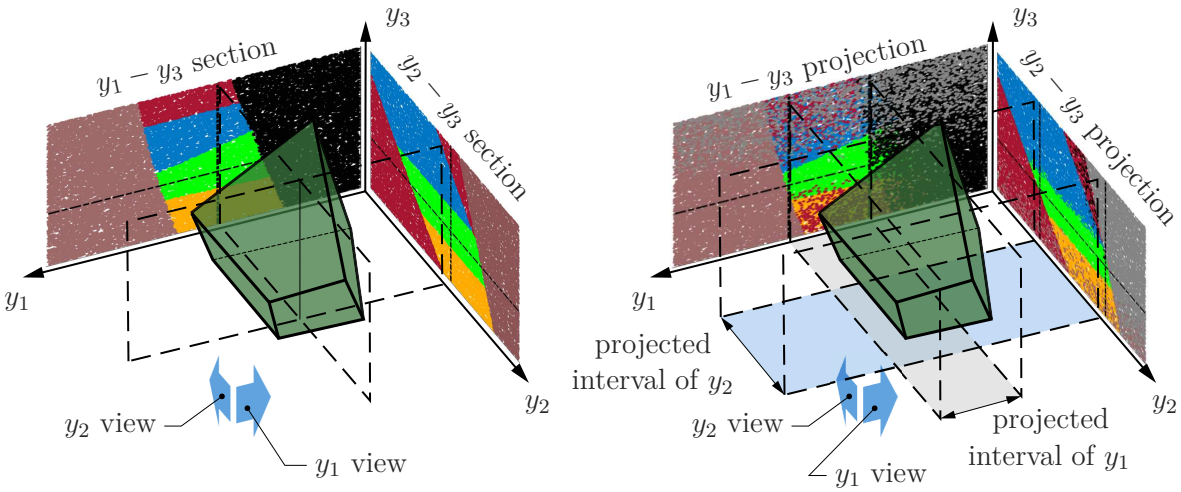
$$\begin{aligned}
&\underset{\mathbf{y}^{\mathrm{lb}},\mathbf{y}^{\mathrm{ub}}}{\text{minimize}} \quad \varphi_{\mathrm{sb}}(\Omega_{\mathrm{sb}}) \\
&\text{with} \qquad \varphi_{\mathrm{sb}} = \begin{cases} - \quad \mu_{\mathrm{sb}} & \text{if } B_{\%} < B_{\%,\mathrm{crit}} \\ - \quad \mu_{\mathrm{sb}} + \mu_{\mathrm{sb}}\left(B_{\%} - B_{\%,\mathrm{crit}}\right) & \text{if } B_{\%} \geq B_{\%,\mathrm{crit}} \end{cases} \\
&\text{subject to} \quad \mathbf{y}^{\mathrm{lb}} \in \boldsymbol{I}_{\mathrm{ds}}^{\mathrm{lb}},\ \mathbf{y}^{\mathrm{ub}} \in \boldsymbol{I}_{\mathrm{ds}}^{\mathrm{ub}}
\end{aligned} \tag{2.19}$$

2.3.5 Visualization of Solution Spaces

Understanding the solution space may be difficult, as especially high-dimensional solution spaces tend to become complex structures. Therefore, in order to get a better understanding, an overview of different visualizations of the solution space is given.

Two Dimensional Solution Spaces consider two design variables. Figure 2.16 (a) presents three system responses (z_1, z_2, z_3) and two design variables (y_1, y_2). Note the existence of the two lower bounds z_2^{lb}, z_3^{lb} which result in system failure (bad design) when they are not met and one upper bound z_1^{ub} that leads to failure if exceeded. Figure 2.16 (b) separates the design space (input space) into a good and bad region, while the good region is the so called solution space.

Two Dimensional Projections of Solution Spaces may be used for solution spaces with more than three dimensions. This Paragraph presents the transition from a two dimensional solution space to a multi dimensional one. The reason for the use of projections instead of sections becomes clear when observing Figure 2.19 (a). By using multiple sections through the solution space, much of information gets lost, as no information is included in the area near the intersection of the section views. This information is increased by the use of projections, see Figure 2.19 (b).

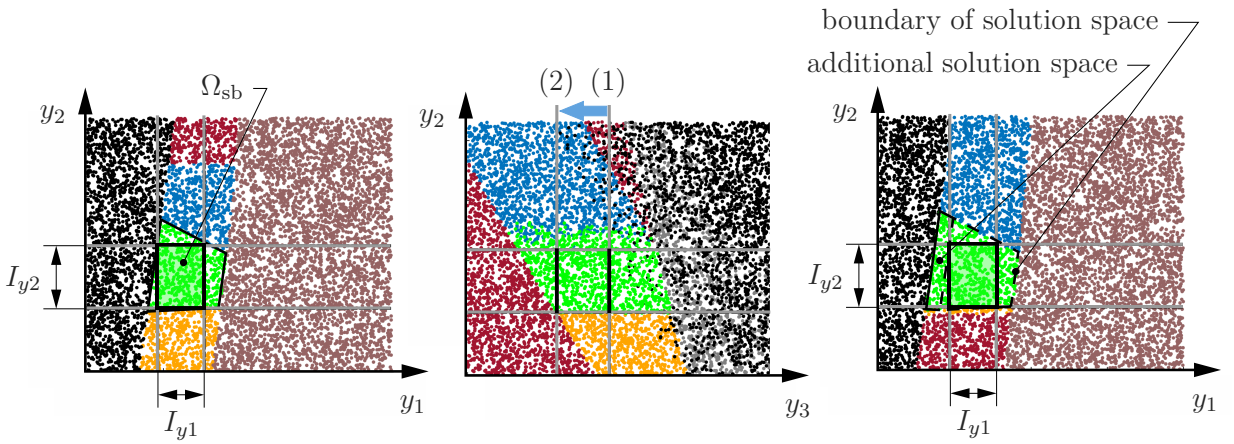


(a) sections through the solution space

(b) projections of the solution space

Figure 2.19: Applying two dimensional projections of high dimensional solution spaces.

Figure 2.20 (a) shows a two dimensional solution space considering five objective criteria. In order to decouple the design variables from each other, a solution box may be laid into the solution space. Since the solution intervals may be forwarded to different departments, they do not have to communicate with each other as long as their solution lies within the required solution interval. In order to maximize the robustness of the final solution and increase feasibility in their integration, the box size measure of the solution box should be maximized $\mu \rightarrow \max$.



(a) $y_1 - y_2$ section view of setup (1)

(b) setup (1) and (2) of y_3

(c) $y_1 - y_2$ section view of setup (2)

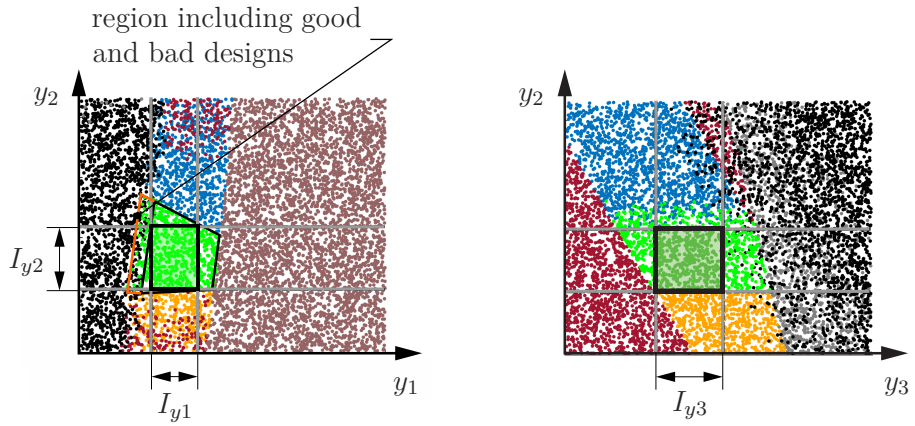
Figure 2.20: Two dimensional sections of high dimensional solution spaces.

Since the vehicle performance does not only depend on two design variables, other design variables also must be considered. Considering y_3 as a third design variable with setup (1), the two dimensional solution space presented in Figure 2.20 (a) presents a two dimensional section through the three dimensional solution space. As y_3 changes from setup (1) to setup (2), see Figure 2.20 (b), the section view $y_1 - y_2$ changes as well. Figure 2.20 (c) makes clear that the adoption of parameter y_3 has a direct influence on the maximum possible two dimensional solution space. Therefore, y_3 can be adapted in order to maximize robustness and flexibility of y_1 and y_2.

If a design interval should be prescribed for y_3 as well, the view on the $y_1 - y_2$ plane no longer represents a section but a projection of the solution interval of y_3 into the $y_1 - y_2$ plane. While in Figure 2.21 (a) the whole design space is shown for two selected design variables (y_1 and y_2), the designs within the solution intervals of all other design variables (y_3) are projected within this plot. Therefore, the user gets an idea in which regions of the design space alternative solutions also exist. Through clever shifting of the solution box boundaries, the systems engineer may find a solution box that fits his needs. Two different approaches for finding a solution box that satisfies the needs of the engineer with respect to feasibility restrictions are presented in Chapter 5.

Statement 3 (sections through a multi-dimensional solution space)
A two dimensional section through a multi-dimensional solution space gives information about the solution space of the two presented design variables. However, this solution space is only valid at the precisely defined value of all other parameters, and therefore unsuitable for the robust and flexible design of more than two design parameters.

Statement 4 (projections through a multi-dimensional solution space)
In a projection view, all designs that lie within the current solution box are projected into the plane of the two presented design variables. This provides full information about the designs within the solution box and also from adjacent areas of good designs.



(a) $y_1 - y_2$ projection view of setup (3)

(b) $y_3 - y_2$ projection view of setup (3)

Figure 2.21: Two dimensional projections of high dimensional solution spaces.

Multi Parameter Plots (MPPs) are an alternative representation of solution spaces with more than two design variables. Figure 2.22 presents a MPP of two different requirement sets (solution boxes) considering three design variables. The first set of requirements is the same as presented in Figure 2.21, and the other is an alternative set that satisfies the overall system requirements. Since MPPs are a very comprehensive way to present solution spaces and also easy to understand, they are well suited for the comparison of different requirement sets. A disadvantage of MPPs is that they can not be adapted manually, since information about alternative areas of good designs is lost.

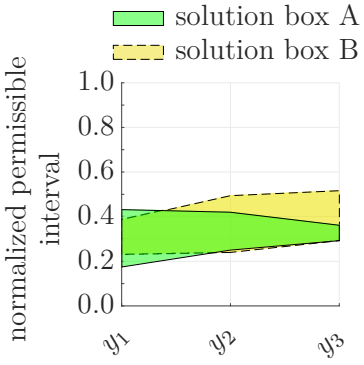
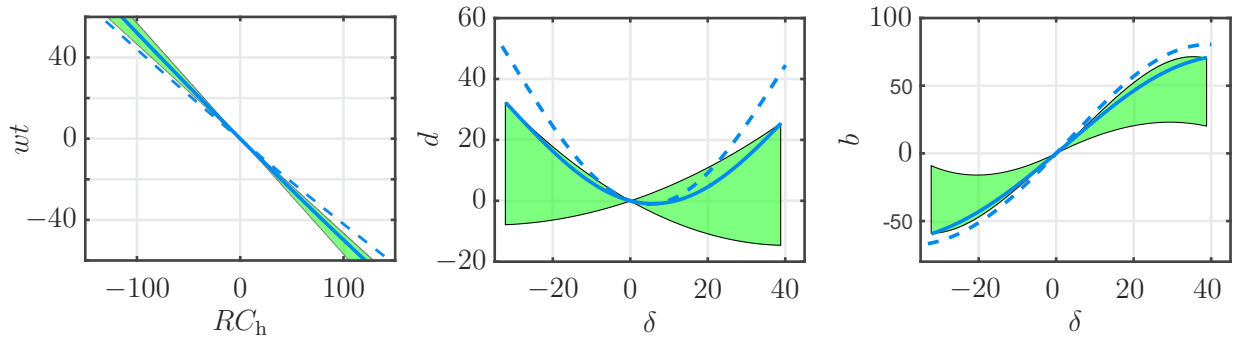


Figure 2.22: MPP of two solution boxes.



(a) one parameter

(b) three parameters

(c) four parameters

Figure 2.23: Solution spaces presented as areas for characteristic curves.

Areas for Characteristic Curves may also represent solution spaces as shown in Figure 2.23. Their major advantage is that development engineers are familiar with characteristic curves, such as kinematic trail over wheel angle, and thus it is easy for them to evaluate and compare them with characteristic curves of existing vehicles and evaluate them by experience. As described for MPPs in the paragraph Multi Parameter Plots (MPPs), they can not be adapted manually, since the information about areas with other possible solutions is lost.

CHAPTER 3

OBJECTIFIED TIRE DEVELOPMENT

In this chapter, the role of the tire within the interlinked design procedure is presented. As it is the only link with the ground, the tire is one of the most relevant subsystems in vehicle design. While the suspension system, aerodynamics, drivetrain, steering system and control systems are there to maximize the use of the tire, the tire itself determines the potential of lateral and longitudinal forces that can be generated in the contact patch. In Figure 3.1, the tire is presented in the overall context of this work, while the dashed line marks the scope of this chapter.

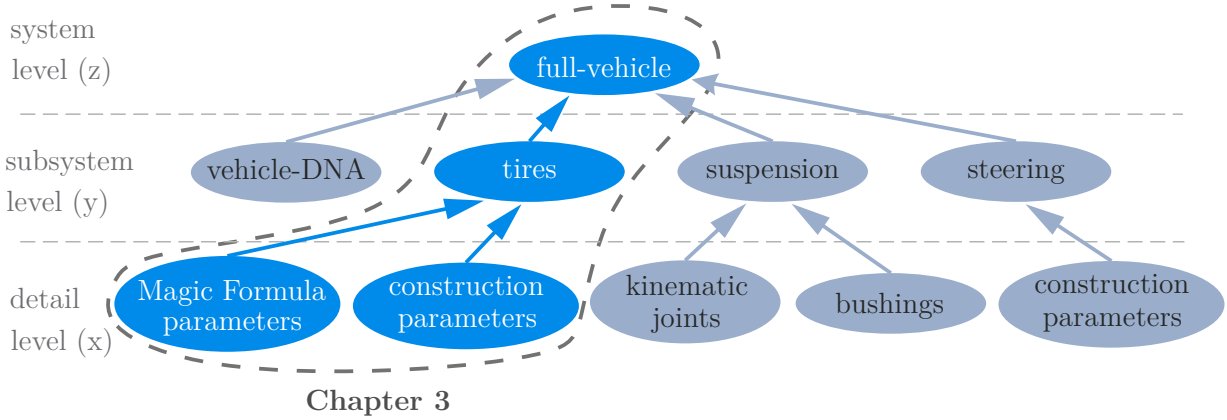


Figure 3.1: The role of tires within the interlinked design procedure.

3.1 Comparison: Classical vs. Modern Tire Development

The motivation for objectified and integrated tire development arises from the shortcomings of the classical approach. Therefore, in this section, the classical development process is reviewed, and weak points are identified. Based on the identified shortcomings, the concept

of the objectified and integrated approach is presented.

3.1.1 Classical Tire Development

Developing a tire in a classical manner means that the tire is evaluated based on the overall vehicle behavior. This behavior may be measured by objectified CVs or, if unavailable, subjectively by a driver, as shown by the a- and b-path in Figure 3.2. Since subjective evaluations by different humans may vary, it can be difficult to find a solution, especially when parties with conflicting goals are involved. No matter whether objective or subjective vehicle measures are used, the behavior of the vehicle not only depends on the properties of the tire but also on many other subsystem properties. In correlation to the dependency on other subsystems such as the suspension system, the tire manufacturer may not be able to achieve the goals imposed on him in each case. Therefore, the performance of the future tire combined with the vehicle of tomorrow can only be estimated based on the knowledge of current vehicles. This results in iteration loops between both parties, where requirements are detailed further and trade-offs evaluated. If the OEM provides target values for particular tire properties, illustrated as c-path in Figure 3.2, they often are not achievable as a whole due to conflicting goals such as rolling resistance and grip. Since the relation between tire properties and overall vehicle targets are blurry, deviations are difficult to evaluate. Therefore, target values for tire properties may only be used as a coarse development trend. Figure 3.3 illustrates the process of classical tire development. Beginning with the development start, the OEM designs the concept of the future vehicle. In parallel, the tire manufacturer starts to develop a new tire based on information from the predecessor. At the end of iteration loop 1, the first prototype is produced. Therefore, the most developed tire at this point is integrated in the vehicle. Following an evaluation of the tire on system level, results are fed back to the tire manufacturer for further development in the following iteration loop 2.

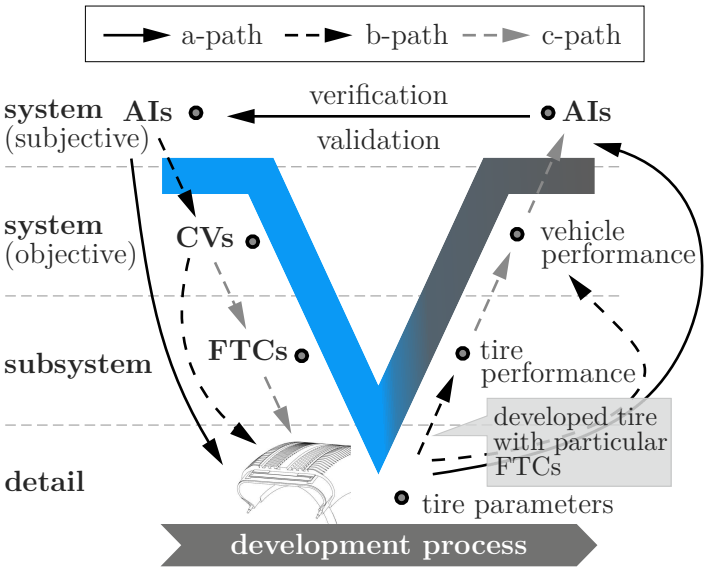


Figure 3.2: Classical tire development in the V-model.

Why is classical tire development critical for future vehicles? As a tire manufacturer has less information about the vehicle of tomorrow than an OEM, it is difficult for him to design a tire that satisfies the overall vehicle requirements by interacting with all other subsystems of the vehicle. Therefore, his ability to build a good tire is limited, if he does not receive exact requirements on the tire performance itself. As a result, multiple iteration

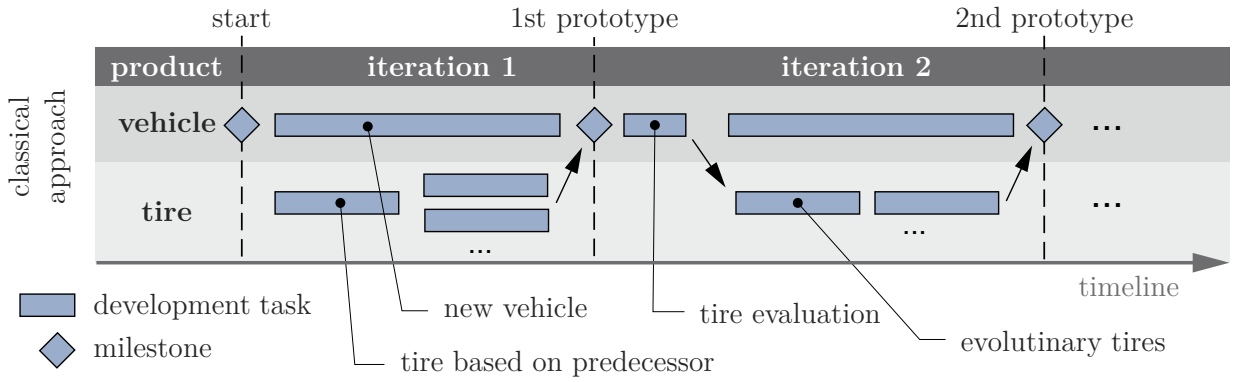


Figure 3.3: The classical tire development process.

loops are necessary in order to consolidate the requirements and integration. This process may repeat until both parties are satisfied with the achieved tire properties. In practice, however, it may be possible that the continuing timeline leads to unintended compromises. Since the tire must first be designed, produced, tested on a test bench and finally delivered to the OEM before its performance can be evaluated, the iteration loops are time expensive compared with iteration loops in the digital industry. Therefore, it is important to use iteration loops wisely in order to reach a high degree of maturity early in the development process. After each iteration loop, the tires delivered by the tire manufacturer are tested on the vehicle by trained drivers. Due to an increasing variety of vehicles, the test and validation expenses will increase a great deal in the future.

3.1.2 Objectified and Integrated Tire Development

The objectified and integrated tire development proposed in this work was developed in order to overcome the shortcomings of the classical approach. The method is integrated into an overall vehicle design strategy. Also, if the design of other subsystems is already fixed, the method is suitable for the pure design of tires. In order to reduce the number of necessary iteration loops, an interface was created that allows the OEM to concentrate on his core competencies, i.e. the system design of the vehicle on one side, and on the other side, it also allows the tire manufacturer to concentrate soley on tire design. Great importance was

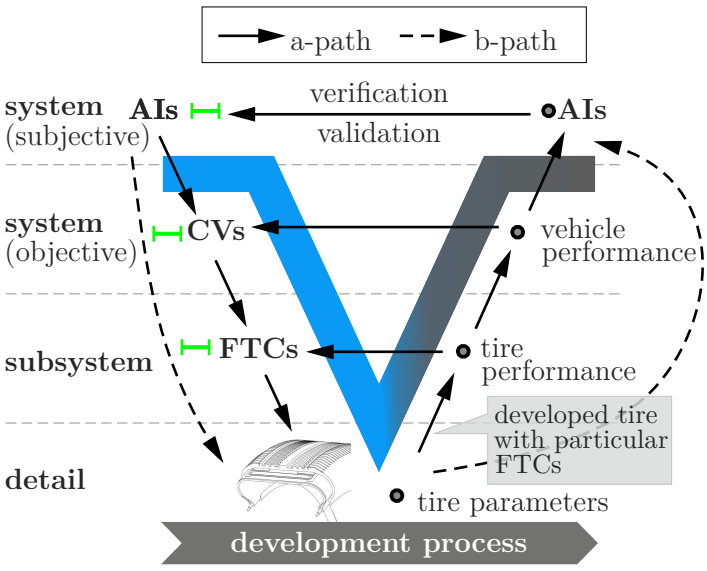


Figure 3.4: Objectified and integrated tire development in the V-model.

also attached to define the interface and information to be exchanged between OEM and tire manufacturer SMART, which stands for Specific, Measurable, Accepted, Reasonable and Time-bound. Therefore, particular *functional tire characteristics* (FTCs) are introduced in Section 3.2. The intent of those FTCs is to reduce the partially subjective and objective evaluation of the tire on the system level by an objective evaluation on the subsystem level. In order to do so, they should clearly describe the influence of the tire on the overall vehicle behavior at maximum flexibility for tire detail design. FTCs are specific and measurable, because specific values may be assigned to them after they were measured on a test bench, for example. In order to use them in a project, they need to be accepted by the parties involved.

Furthermore, since FTCs are independent of other subsystems in the vehicle, they can be evaluated more easily by the tire manufacturer, if the requirements derived by the OEM are reasonable. Additionally, the requirements on the FTCs must be satisfied at a specific milestone in the project and are therefore time-bound. As part of an overall design strategy, based on objectified overall vehicle targets, requirements on FTCs are derived as permissible intervals. Any tire that fulfills the requirements imposed on its functional properties is a good one and helps to satisfy the overall vehicle targets. In order to avoid infeasibility of the requirements derived on the tire, only feasible tires are considered during the search for a permissible interval.

Figure 3.4 presents the described process integrated into the V-model. Due to the permissible ranges for the different abstraction levels, whether a tire might be good or not can be evaluated earlier in the development process. If an *Assessment Index* (AI) is not already objectified, it still needs to be considered by subjective evaluations of the vehicle. For OEMs and tire manufacturers, deriving requirements as permissible intervals on FTCs offers a great opportunity for saving both development time and cost. Still, it is associated with two specific challenges: First, building an appropriate quantitative bottom-up simulation model, and second, to check and enable the feasibility of the subsystem requirements (y-level) derived from the system (z-level) on the detail level (x-level). In the case of the tire subsystem, it is necessary to distinguish between a virtual and physical parameterization on the detail level, which will be described in detail in Subsection 3.2.2.

3.2 Functional Tire Characteristics (FTCs)

In order to increase the efficiency of the collaboration between OEM and tire supplier, it is crucial for the supplier to obtain a clear set of specifications on the component that should be delivered by them. Today, the performance of the tire is evaluated on a system level by a test driver. Although these drivers are trained to evaluate the vehicle behavior, complete objectivity cannot be fully guaranteed due to differences in their skills, senses, and expectations. Therefore, it could happen that a particular tire is evaluated as a bad tire, although it already has gone through a number of iteration loops on the supplier side.

The target of the integrated and objectified design approach is to reduce those iteration loops. This is done by dividing the design process of the tire into three different abstraction

levels. The abstraction levels are divided into a system, subsystem and detail level. While the OEM has its core competency in how the overall product should look and feel and which requirements are therefore necessary (system design), the tire manufacturer knows best how to realize those subsystem requirements by tuning his component on a detail level (subsystem design). Therefore, in the case of tires, FTCs are functional properties of the tire on the subsystem level and build a bridge between OEM and tire manufacturer. In (Niedermeier et al. 2013), the vehicle dynamics behavior was optimized by the adoption of FTCs, called tire CVs there. However, the presented example only considers a few of the possible tire properties, which is not appropriate for the method presented in this work. In addition, the selection of the relevant FTCs strongly depends on the objective criteria considered for the overall vehicle behavior. Therefore, the required set of FTCs is determined separately for this work in Subsection 3.2.1, before a new method for the calculation of vehicle CVs based on FTCs is introduced in Section 3.3.

3.2.1 Most Influential FTCs

From a functional point of view, a tire can be represented by multiple FTCs. In (Niedermeier et al. 2013), for example, it has been said that the whole set of tire properties consists of 60 values. Possible FTCs of the lateral/longitudinal force over slip angle and aligning torque over slip angle are the:

- Offset and gradient at zero slip angle
- Coordinates of the grip peak (slip angle and maximum grip)
- Coordinates at the maximum slip angle (slip angle and grip)

In order to improve the efficiency of the method without neglecting accuracy, the most influential FTCs are selected in this subsection. The aim is to identify as many FTCs as necessary, but also as few as possible, in order to describe the tire behavior.

Therefore, in a first step, the most influential FTCs were identified by computing the total-effect Sobol Indices between potential FTCs and objective criteria of the vehicle, in order to analyze their sensitivity. Sobol indices are intended to represent the sensitivities for general nonlinear models (Eichstetter et al. 2013) and belong to the group of variance-based sensitivity analysis. They represent the influence of a design parameter on the variance of the objective criteria. Compared to the first-order Sobol Indice, the total-effect Sobol Indice also considers the variance caused in the output space by the interaction of the regarded design variable with all other design variables.

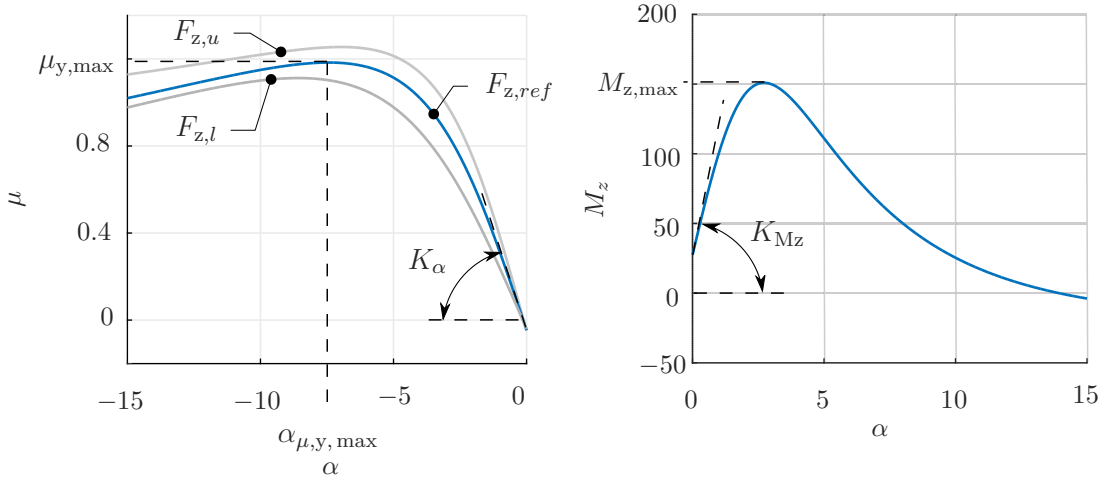
In a second step, a correlation analysis between the FTCs, which were identified as the most influential ones, leads to a reduced set of FTCs.

As a result, the potential FTCs: μ_{15° and $\alpha_{\mathrm{Mz,\,max}}$ are not considered as design variables, since their value is already indirectly specified by specifying the values of the FTCs considered within the reduced set. Figure 3.5 presents the reduced set of FTCs considered in this work. The grip degression over the vertical tire load cannot be depicted in those

graphs but is calculated by

$$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}} = -\frac{\mu_{\mathrm{y,\,max}}(F_{z,u}) + \mu_{\mathrm{y,\,max}}(F_{z,l})}{2\,\mu_{\mathrm{y,\,max}}(F_{z,\mathrm{ref}})} \tag{3.1}$$

where $F_{\mathrm{z,ref}}$ represents a reference vertical load on the tire, while $F_{\mathrm{z,u}}$ and $F_{\mathrm{z,l}}$ are vertical loads above and below the reference load with the same distance to it (e.g., $\pm 1\,\mathrm{kN}$).



definition	symbol	range	unit
max. lateral grip coefficient	$\mu_{\mathrm{y,max}}$		-
lateral grip degression	$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$		1/N
cornering stiffness	K_α		N/°
slip angle at max. lateral grip	$\alpha_{\mu,\mathrm{y,\,max}}$		°
max. self-aligning torque	$M_{\mathrm{z,\,max}}$		Nm
self-aligning torque stiffness	K_{Mz}		Nm/°
relaxation length	σ_{y0}		m
vertical stiffness	K_{z}		N/mm

Figure 3.5: Definition of the FTCs.

In Table 3.1, the total-effect Sobol Indices for the eight relevant FTCs considered for this work are presented for selected objective criteria. As a result, the design engineer can see which FTC he needs to adapt in order to influence a particular vehicle CV, or if an FTC cannot be adapted, which alternative design variables exist. When viewing a particular column in Table 3.1, it can be seen which vehicle CVs also change when adapting a particular FTC. It gets clear that for the considered objective criteria, which mainly evaluate lateral dynamics, no requirements are derived on the vertical stiffness of the tire K_{z}. Furthermore, the cornering stiffness K_α is the most influential FTC in terms of the number of objective criteria influenced by (14 CVs), and its impact ($S_{\mathrm{T},T_{\mathrm{eq}}} = 0.97$). Therefore, 97 % of the variation of the time delay between steering input and yaw velocity T_{eq} results from the variation of K_α and its interaction with the other design variables.

Table 3.1: Influence of FTCs on objective criteria measured by Sobol Indices.

	$\mu_{\mathrm{y,\,max}}$		$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$		K_α		$\alpha_{\mu,\mathrm{y,\,max}}$		$M_{\mathrm{z,\,max}}$		K_{Mz}		σ_{y0}		K_{z}	
lb DS	1.01		0.95		-2200 Nm		-14.7 °		-295 Nm		21 $^{\mathrm{Nm}}/_{\circ}$		0.51 m		260 $\frac{N}{\mathrm{mm}}$, 0.95	
ub DS	1.43		0.99		-1400 Nm		-5.3 °		-12 Nm		181 $^{\mathrm{Nm}}/_{\circ}$		0.77 m		350 $\frac{N}{\mathrm{mm}}$, 1.05	
	FA	RA	FA	RA	FA	RA	FA	RA	FA	RA	FA	RA	FA	RA	FA	$^{FA}/_{RA}$
$\Delta\delta_{\mathrm{h}}\,(7\,\mathrm{m/s^2})$	0.25	0.05	0.00	0.00	0.26	0.22	0.11	0.03	0.04	0.01	0.01	0.00	0.00	0.00	0.00	0.00
$\Delta\delta_{\mathrm{h,95\%}}$	0.33	0.20	0.01	0.00	0.01	0.00	0.14	0.04	0.01	0.01	0.03	0.01	0.00	0.00	0.00	0.00
$\Delta\beta\,(7\,\mathrm{m/s^2})$	0.00	0.16	0.00	0.00	0.00	0.71	0.00	0.09	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
$K\,(4\,\mathrm{m/s^2})$	0.01	0.00	0.00	0.00	0.56	0.37	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
$\dot{\psi}/_{\delta_{\mathrm{h}}\mathrm{max}}$	0.01	0.00	0.00	0.00	0.45	0.46	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
$\dot{\psi}/_{\delta_{\mathrm{h}}190}$	0.01	0.00	0.00	0.00	0.43	0.49	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00
$F_{\mathrm{z,\,min}}$	0.88	0.01	0.06	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$a_{\mathrm{y,\,max}}$	0.88	0.01	0.06	0.00	0.01	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
$gM_{\mathrm{h}}\,(1\,\mathrm{m/s^2})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.00	0.00	0.00
$\Delta M_{\mathrm{h,loss}}$	0.04	0.05	0.00	0.01	0.07	0.02	0.03	0.02	0.14	0.00	0.54	0.00	0.00	0.00	0.00	0.00
$M_{\mathrm{h,\,max}}$	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.82	0.00	0.16	0.00	0.00	0.00	0.00	0.00
$\dot{\psi}/_{\delta_{\mathrm{norm}}\mathrm{max}}$	0.03	0.01	0.00	0.00	0.38	0.48	0.02	0.01	0.01	0.00	0.00	0.01	0.00	0.06	0.00	0.00
T_{eq}	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00
$\angle\varphi^{\circ}$	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.01	0.00
$\delta_{\mathrm{h,f,swd}}$	0.23	0.41	0.01	0.01	0.00	0.01	0.05	0.01	0.01	0.00	0.02	0.02	0.00	0.00	0.00	0.00

if $S_{\mathrm{T},j} \geq 0.05$

Nevertheless, the strength of the influence also depends on the design space considered for the calculation of the Sobol Indices, which must be considered when making decisions.

3.2.2 Causal Relationships of FTCs

The qualitative causal relationships can be represented by dependency graphs in order to get a fast and intuitive overview of them. Therefore, Figure 3.6 presents the causal relationships associated with the tire subsystem.

The relations between system and subsystem level are set based on the Sobol Indices provided in Subsection 3.2.1 by Table 3.1. If the Sobol Indice $S_{T,j}$ between a particular FTC and vehicle CV is greater or equal 0.05 (blue fields in Table 3.1), it is assumed that the interaction between both is relevant. Therefore, those properties are connected by a line within the dependency graph. Depending on the threshold value of the Sobol Indices, the depth of information provided by the dependency graph varies. In case a greater threshold value is selected, the dependency graph would more likely show the major influences, which increases clarity. If the threshold value is lower, weaker influences are shown as well, which can help to find the root cause of a specific design problem. Based on a correlation analysis, it is distinguished between relations that are directly proportional and inversely proportional. Therefore, the engineer gets the information in which direction he needs to adapt a design variable in order to adjust the system performance in the way he wants.

Between the subsystem and detail level, two different parameterizations are used in this work. One parameterization is based on the design variables of the MF-Tire model and the other on construction parameters of the tire as shown at the beginning of this chapter in Figure 3.1. While the MF-Tire model is necessary to create a new model between the subsystem properties of the tire and overall vehicle CVs, which will be explained in Section 3.3, the construction parameters are those which may be adjusted by the tire manufacturer in order to fulfill the requirements on the subsystem. Therefore, the construction parameters are shown on the detail level in the dependency graph, since they are physical and independent of a specific tire model. The information about the presented dependencies between subsystem and detail level properties are based on (Schramm et al. 2017). If no relationship exists between a particular FTC and at least one of the properties on the detail level, then no information is available in current literature. However, it may be assumed that variables on the detail-level which influence K_α also influence $\alpha_{\mu,\mathrm{y,\,max}}$ since a change at the beginning of the lateral slip curve also affects the rest of it. Although the design parameters on the detail level, such as the design of the chafer or apex etc., are the major lever to influence the subsystem performance, in some cases the OEM sets constraints to them as well, such as *overall diameter* (OD) or *overall width* (OW). The underlying reasons are, e.g., optical design/styling requirements or other requirements that can not be specified on the subsystem level yet. The graph can guide the engineer along different variables, if he wants to make a change in the system behavior or trade-offs between different design variables.

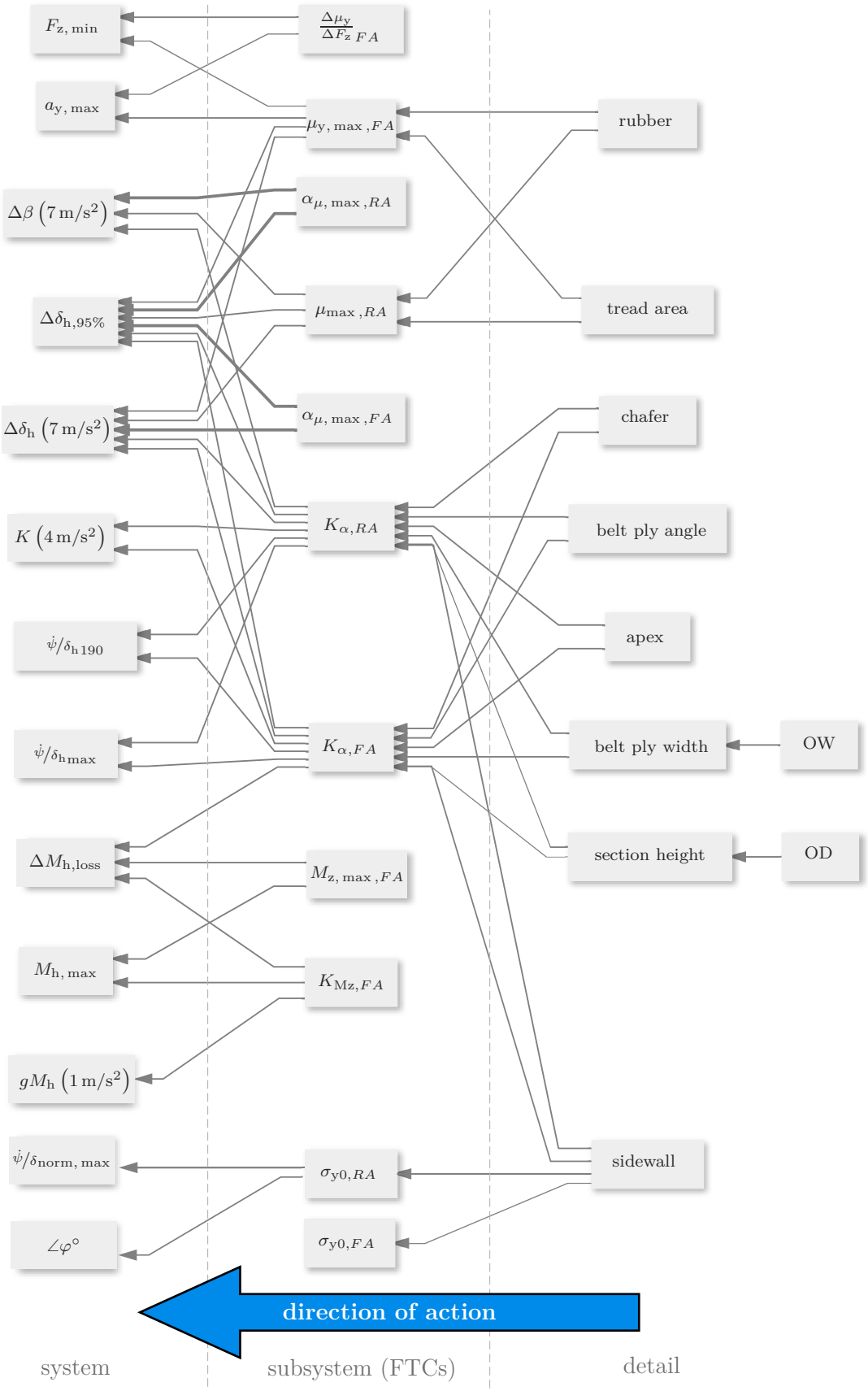


Figure 3.6: Dependency graph of the tires.

3.3 Modeling of the Quantitative Dependencies between FTCs and vehicle CVs

Many publications discuss the parameterization and optimization of various tire models. The MF-tire model is possibly the most popular one so far, especially when it comes to vehicle dynamics simulation. In order to improve the interaction between the tire and the rest of the vehicle, MF-parameters can be optimized together with other design variables of the vehicle, e.g., see (Na and Gil 2016). The tire manufacturer can be supplied with the tire model and try to produce a tire that matches the characteristic curves of the model. Newer publications such as (Niedermeier, Peckelsen, and Gauterin 2013) and (Niedermeier et al. 2013) point to nested optimizations, where the overall vehicle behavior is optimized by adapting FTCs which are achieved by adapting MF-Parameter. The advantage of this approach is that those FTCs that are important for the system design are defined by the OEM and directly optimized. However, for complex design tasks, this approach seems to be time expensive due to the large number of MF-tire model optimizations, and therefore not suitable to derive permissible intervals on the FTCs. In addition, the result of all these approaches are particular values for several tire properties, which is the reason for even stronger conflicting goals as described in Subsection 3.1.1.

In order to enable objectified and integrated tire development, a new method to calculate the vehicle CVs based on FTCs is presented. The method uses two different supervised learning techniques: On one side, regression models calculate vehicle CVs based on a set of FTCs, and on the other side classification is used to distinguish between a feasible and non-feasible set of FTCs. Both functionalities are necessary for the success of the method and will be presented in this section.

3.3.1 Computing CVs based on FTCs

One of the big challenges in deriving requirements on subsystem properties of the tire as permissible intervals is that no appropriate quantitative model between FTCs and vehicle CVs exists yet. Therefore, first, a new model is created. The new tire model must satisfy the following requirements:

- Calculate the vehicle CVs based on a defined set of FTCs such that $CV = f_1(FTC)$.
- Low computational effort.
- Sufficient accuracy in order to make design decisions that influence the series production of major OEMs.

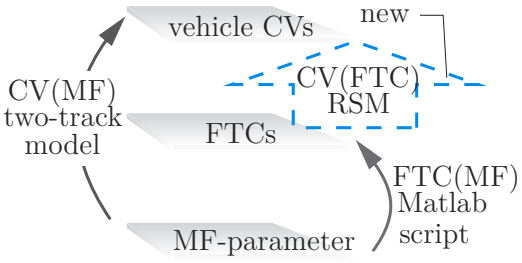


Figure 3.7: Quantitative simulation models including tire properties.

In this work, the new model that satisfies all of these requirements is built by means of Machine Learning. Therefore, two already existing simulation models are used to generate the data necessary for the learning process. Both models include MF-parameters as design variables which can be varied. One model is a complex two-track model that calculates the vehicle CVs based on MF-parameters such that CV = f_1^*(MF). In addition, the two-track model can also include many other design variables from different subsystems such as the suspension system, steering system and vehicle-DNA such that CV = f_1^*(MF, ...). The other model is a Matlab® script that calculates the FTCs based on the MF-tire model, virtual experiments, and calculations such that the FTCs are given by FTC = f_2(MF). Based on the data provided by both models, an RSM is trained to provide the vehicle CVs based on FTCs such that CV = f_1(FTC) as well as other functional subsystem properties such that CV = f_1(FTC, ...).

How to Generate the Model?

First, a DoE of the MF-parameters is done. Therefore, Monte Carlo Sampling is used, which is a stochastic method that allows varying defined variables in a defined design interval randomly. Due to the law of large numbers, the distribution of the sample points can be considered as evenly distributed, if the number of sample points is large enough (Graham and Talay 2013; *The Law of Large Numbers and the Monte Carlo Method* 2014). Which particular MF-parameters need to be varied depends on the FTCs that should be considered. Based on the selection of FTCs presented in Subsection 3.2.1, the MF-parameters shown in Table 3.2 are varied in order to obtain a sampling of FTCs where each variable is adjustable as independently as possible. In a second step, the corresponding FTCs are calculated by a Matlab script for each sample point. Since the relations between MF-Parameters and FTCs are non-linear, the output of the originally evenly distributed MF-sampling is not evenly distributed on the FTC level anymore. Nevertheless, an evenly

Table 3.2: MF-Parameter according to Pacejka (2006) that are varied by Monte Carlo sampling in step 1.

MF-param.	description / effect on
$\lambda_{\mathrm{Ky}\alpha}$	scale factor of the cornering stiffness K_α
p_{Dy1}	scale factor of the lateral friction $\mu_{\mathrm{y,max}}$
p_{Dy2}	variation of friction $\mu_{\mathrm{y,max}}$ with load
p_{Cy1}	shape factor for lateral forces
$\lambda_{\sigma\alpha}$	scale factor of the Relaxation length of F_y
λ_t	scale factor of the Peak of pneumatic trail
q_{Dz1}	peak trail Dpt" = Dpt $\cdot$ ($F_\mathrm{z}/F_{\mathrm{z0}} \cdot R_0$)
q_{Dz2}	variation of the peak Dpt" with load
q_{Cz1}	shape factor Cpt for pneumatic trail
q_{Bz1}	trail slope factor for the trail Bpt at F_{z0}
q_{Bz9}	slope factor Br of the residual torque M_{zr}
q_{Bz10}	slope factor Br of the residual torque M_{zr}
q_{Ez1}	trail curvature Ept at F_{z0}

distributed sampling would offer the possibility to use less sample points for the machine learning procedure between FTC and vehicle CV level in order to obtain an RSM with a good accuracy. In addition, the more input and output variables that need to be considered by the RSM, the greater the number of required sample points, which increases the training time for the model as well. Therefore, the number of sample points should not be larger than necessary. In step 3, a sample point filter is applied on the FTC level in order to generate an evenly distributed sampling. For the exclusion of sample points, the design space on the FTC level is divided into several multidimensional cuboids with the same size. If the number of sample points within a particular cuboid exceeds a certain threshold value, the superfluous ones are deleted. The specific sample point within the cuboid that will be deleted is decided randomly. This approach requires that more sample points than actually necessary for the training of the RSM between FTC and vehicle CV level were created during the DoE on the MF level. Since the Matlab script is very inexpensive in terms of calculation time, multiple runs are no problem. After the sample points are filtered, in step 4 the corresponding MF-parameter sets are forwarded to the two-track model for experimental procedure and post-processing. As a result of step 4, an evenly distributed FTC sampling with corresponding vehicle CVs becomes available. Based on the generated data from the previous steps, Machine Learning is used to create an RSM that fits best to the relation between input and output data in the final step. Therefore, ANNs are trained using the Neural Network Toolbox in Matlab®. The objective of the training is to maximize the R^2 value of the validation data using an optimization. The resulting neural network may be used to calculate the vehicle CVs based on FTCs in a very efficient and accurate manner by considering all of the details that were implemented into the both already existing models.

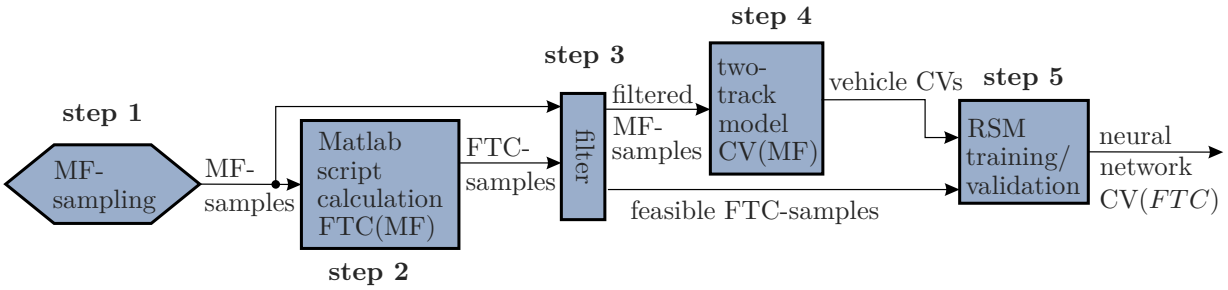


Figure 3.8: Workflow to create an RSM that calculates the vehicle CVs based on FTCs.

3.3.2 Identifying Non-Feasible Designs

The parameterization of the tire by use of physical properties, the so-called FTCs as described in Subsection 3.3.1, leads to the following problem: As they cannot be varied completely independently from each other, there are certain regions within the design space where no sample points exist, and therefore no system response based on a physical model can be calculated. Therefore, the neural network is trained without considering those regions. Measured data of existing tires also confirm this correlation. The ultimate

result is that the neural network needs to extrapolate the performance, if the design point that should be computed lies outside of the feasible space, which is very inaccurate. Table 3.3 compares the correlation coefficient of the existing tire portfolio and the correlation of the DoE procedure described in Subsection 3.3.1. While the FTCs of the existing tires are strongly correlated, the correlation of the virtual DoE samples is more relaxed. The data shows that $\mu_{\mathrm{y,\,max}}$ of the existing tires has a strong correlation with K_α, $\alpha_{\mu,\mathrm{y,\,max}}$ and K_{Mz}. However, differences can be identified compared to the sample points of the virtual tires. The correlation with K_α does not exist for virtual tires, which may be correct as the rubber and tread area of the tire are the main design variables for $\mu_{\mathrm{y,\,max}}$ and the sidewall design for K_α. For the correlation with $\alpha_{\mu,\mathrm{y,\,max}}$, the factor has the opposite sign compared to existing tires. The correlation with K_{Mz} for existing tires makes sense, since both depend on similar detail parameters such as the rubber or tread area. Nevertheless, it is not considered for the virtual tires, as the MF-tire model is able to represent those characteristics. The correlation between $\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$ and K_α is not represented for the virtual tires either. In addition, a strong correlation between $M_{\mathrm{z,\,max}}$ and K_{Mz} exists for the data of existing tires as well as for the virtual tires.

Table 3.3: Correlation between the FTCs.

existing tires	$\mu_{\mathrm{y,\,max}}$	$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$	$\alpha_{\mu,\mathrm{y,\,max}}$	K_α	$M_{\mathrm{z,\,max}}$	K_{Mz}	σ_{y0}
$\mu_{\mathrm{y,\,max}}$	1.00	0.25	**0.37**	-0.56	-0.23	0.32	-0.09
$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$	0.25	1.00	0.15	-0.44	0.13	0.02	-0.20
$\alpha_{\mu,\mathrm{y,\,max}}$	**0.37**	0.15	1.00	-0.12	-0.46	0.52	0.07
K_α	-0.56	-0.44	-0.12	1.00	0.03	-0.30	0.22
$M_{\mathrm{z,\,max}}$	-0.23	0.13	-0.46	0.03	1.00	-0.91	-0.47
K_{Mz}	0.32	0.02	0.52	-0.30	-0.91	1.00	0.35
σ_{y0}	-0.09	-0.20	0.07	0.22	-0.47	0.35	1.00

DoE	$\mu_{\mathrm{y,\,max}}$	$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$	$\alpha_{\mu,\mathrm{y,\,max}}$	K_α	$M_{\mathrm{z,\,max}}$	K_{Mz}	σ_{y0}
$\mu_{\mathrm{y,\,max}}$	1.00	0.29	**-0.23**	-0.00	-0.06	-0.02	0.00
$\frac{\Delta\mu_{\mathrm{y}}}{\Delta F_{\mathrm{z}}}$	0.29	1.00	-0.04	0.00	-0.02	-0.00	-0.00
$\alpha_{\mu,\mathrm{y,\,max}}$	**-0.23**	-0.04	1.00	-0.27	-0.01	0.06	0.00
K_α	-0.00	0.00	-0.27	1.00	0.01	-0.24	-0.00
$M_{\mathrm{z,\,max}}$	-0.06	-0.02	-0.01	0.01	1.00	-0.76	-0.00
K_{Mz}	-0.02	-0.01	0.06	-0.24	-0.76	1.00	0.01
σ_{y0}	0.00	-0.00	0.00	-0.00	-0.00	0.01	1.00

The correlation between $\alpha_{\mu,\mathrm{y,\,max}}$ and K_α can be explained by the physical tire behavior: A tire with high cornering stiffness K_α and low slip angle at maximum grip $\alpha_{\mu,\mathrm{y,\,max}}$ as well as a tire with low cornering stiffness K_α and high slip angle at maximum grip $\alpha_{\mu,\mathrm{y,\,max}}$ are representable. Nevertheless, if a tire with low cornering stiffness K_α and low slip angle at maximum grip $\alpha_{\mu,\mathrm{y,\,max}}$ should be built, this might not be possible depending on the

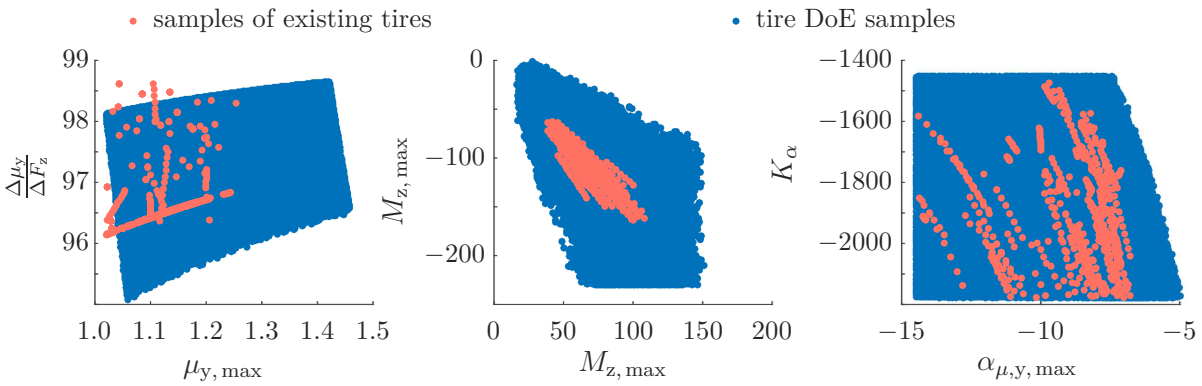


Figure 3.9: Visualization of the correlations between the FTCs regarding the DoE samples and existing tires.

difference between the two and assuming that the maximum lateral grip $\mu_{y,\,max}$ should be the same, see Figure 3.9. If the correct MF-parameters were selected to be varied within an appropriate design interval, it can be assumed that those FTC combinations which cannot be represented by the DoE also cannot be realized by any manufacturer. The reason for this is that the MF-tire model is the usual model for vehicle dynamics investigations of all types of tires. Therefore, the area without sample points is called the *non-feasible space* $\Omega_{\text{non-feas}}$. Figure 3.10 shows projections of the 8-dimensional feasibility space for three selected FTCs. To identify and separate the feasible space from the non-feasible space, *Support Vector Machine* (SVM) methods are used.

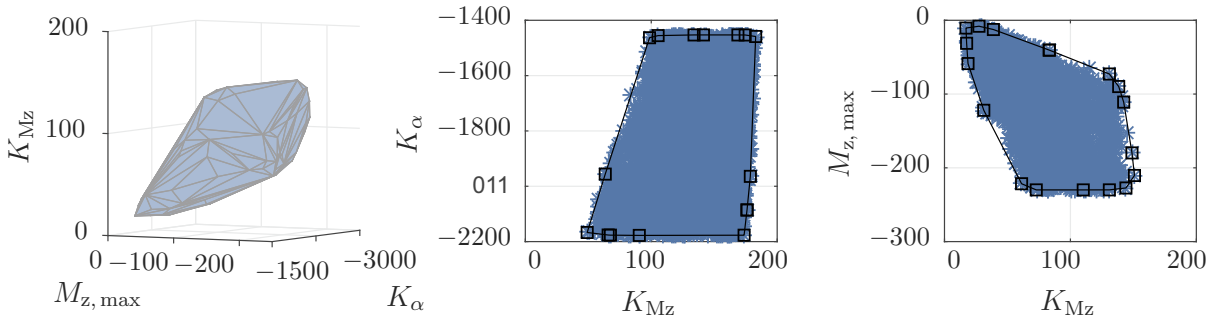
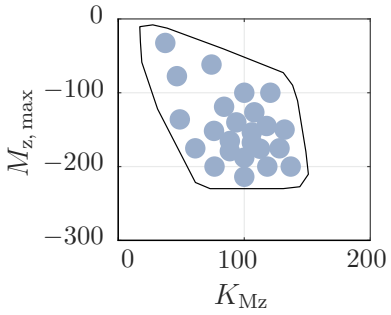
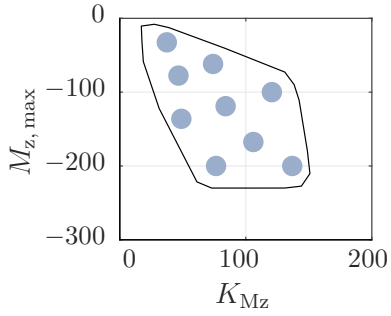
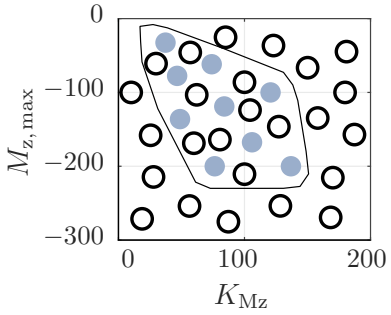


Figure 3.10: Distribution of FTC samples in the feasibility space.

In order to keep the solution box from growing into the non-feasible region of the design space, this region must be labeled accordingly. Classifiers may be used to distinguish the different data sets using supervised machine learning. Common classifiers are Classification Trees, Naive Bayes, Nearest Neighbors, Support Vector Machine Classification (The MathWorks Inc. 2016). The following procedure explains the creation of a binary classifier to distinguish between tires that belong to the feasible or non-feasible space. Therefore, Support Vector Machine Classification is used due to its good applicability to "real world" problems and suitability for high dimensions. The entire procedure consists of eight consecutive steps, which are described in Table 3.4, and

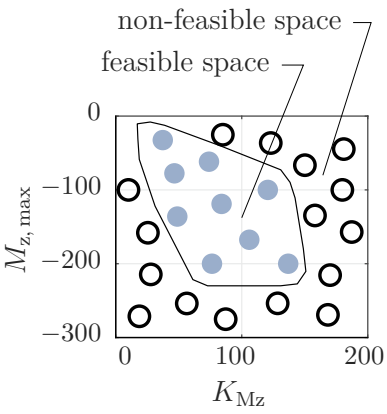
is integrated into the creation process of the quantified model described in Subsection 3.3.1.

Table 3.4: Application of support vector machine (SVM) classification for calculating solution spaces for FTCs.

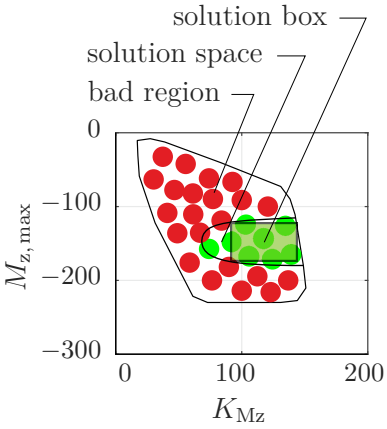
Parameter Space	Description
	**step 1:** The procedure starts with a Monte Carlo sampling of relevant parameters of the MF-tire model. Based on the FTCs presented in Figure 3.5, the MF-parameters shown in Table 3.2 were selected for the Monte Carlo sampling. **step 2:** In step 2, the FTCs are calculated for all sample points by the calculation rule presented in Subsection 3.2.1.
	**step 3:** In step 3, the FTC samples and the associated MF-parameters are filtered in order to get a preferably homogeneous distribution of sample points within the FTC space and to exclude outliers. Afterward, the filtered MF-parameters are forwarded to the DoE of the two-track model on which the neural network for the solution space calculation relies.
	**step 4:** Based on the filtered FTC samples, a second Monte Carlo sampling is performed. Contrary to the Monte Carlo sampling in step 1, this one is done directly within the FTC-space and therefore, the sample points are evenly distributed within the seven-dimensional hyperbox that consists of the seven FTCs presented in Figure 3.5.

Continued on next page

Table 3.4 – continued from previous page.

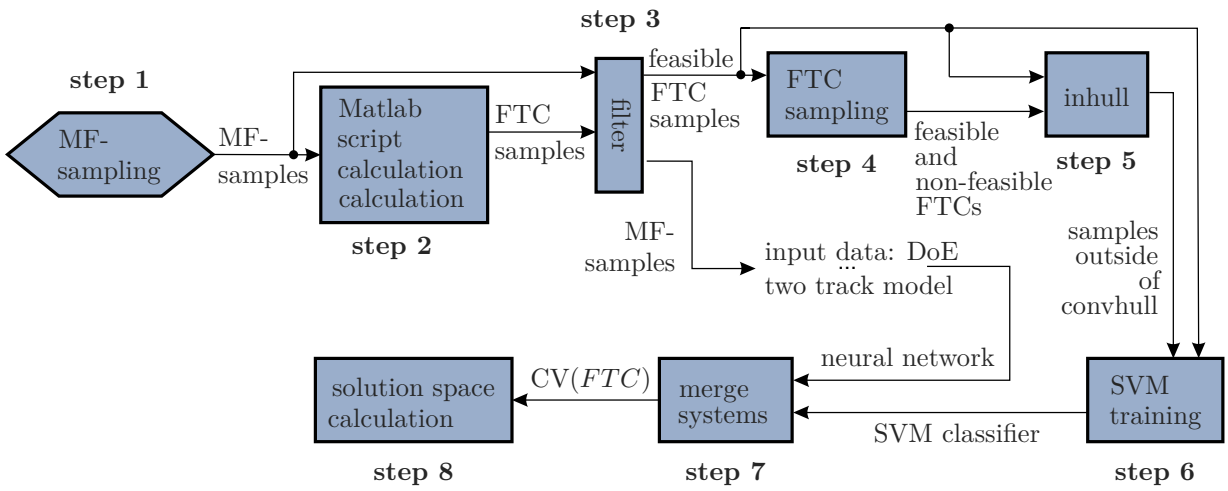


step 5: In step 5, a convex hull that encloses all filtered FTC samples from step 3 is generated. Based on the convex hull, the FTC samples from the second Monte Carlo sampling (step 4) are divided into two groups, the feasible ones inside the convex hull and the non-feasible ones outside of it. Since we already have feasible designs from step 3, the feasible sample points that were created in step 4 are deleted. The sample points outside of the hull are forwarded to step 6 along with the ones on which the hull was created.
step 6: In step 6, a Support Vector Machine is trained to distinguish between feasible and non-feasible FTCs.



step 7: In step 7, the two surrogate models are merged into one system.
step 8: While the support vector machine filters designs that are not feasible on a detail level (x-level), the neural network calculates the system performance of the remaining designs on the z-level. Therefore, it is not permitted to shift the solution box to a region where feasibility is not ensured.

Workflow to identify non-feasible designs during the solution space calculation:



3.4 Feasibility of the Requirements on FTCs

The feasibility of a particular tire that satisfies the requirements on the FTCs is ensured in two steps.
In the first step, the design space is separated into the feasible space Ω_{feas} and non-feasible space $\Omega_{\text{non-feas}}$, as described in Subsection 3.3.2. Designs belonging to the non-feasible space $\boldsymbol{y} \in \Omega_{\text{non-feas}}$ are excluded from the solution box optimization. Therefore, they are first identified by a SVM according to Equation (3.2) and then deleted.

$$z_{\text{feas}}^{\text{FTC}} = \begin{cases} \leq 0 & \text{if } \boldsymbol{y} \in \Omega_{\text{non-feas}} \quad \rightarrow \text{feasible} \\ > 0 & \text{if } \boldsymbol{y} \in \Omega_{\text{feas}} \quad \rightarrow \text{infeasible} \end{cases} \tag{3.2}$$

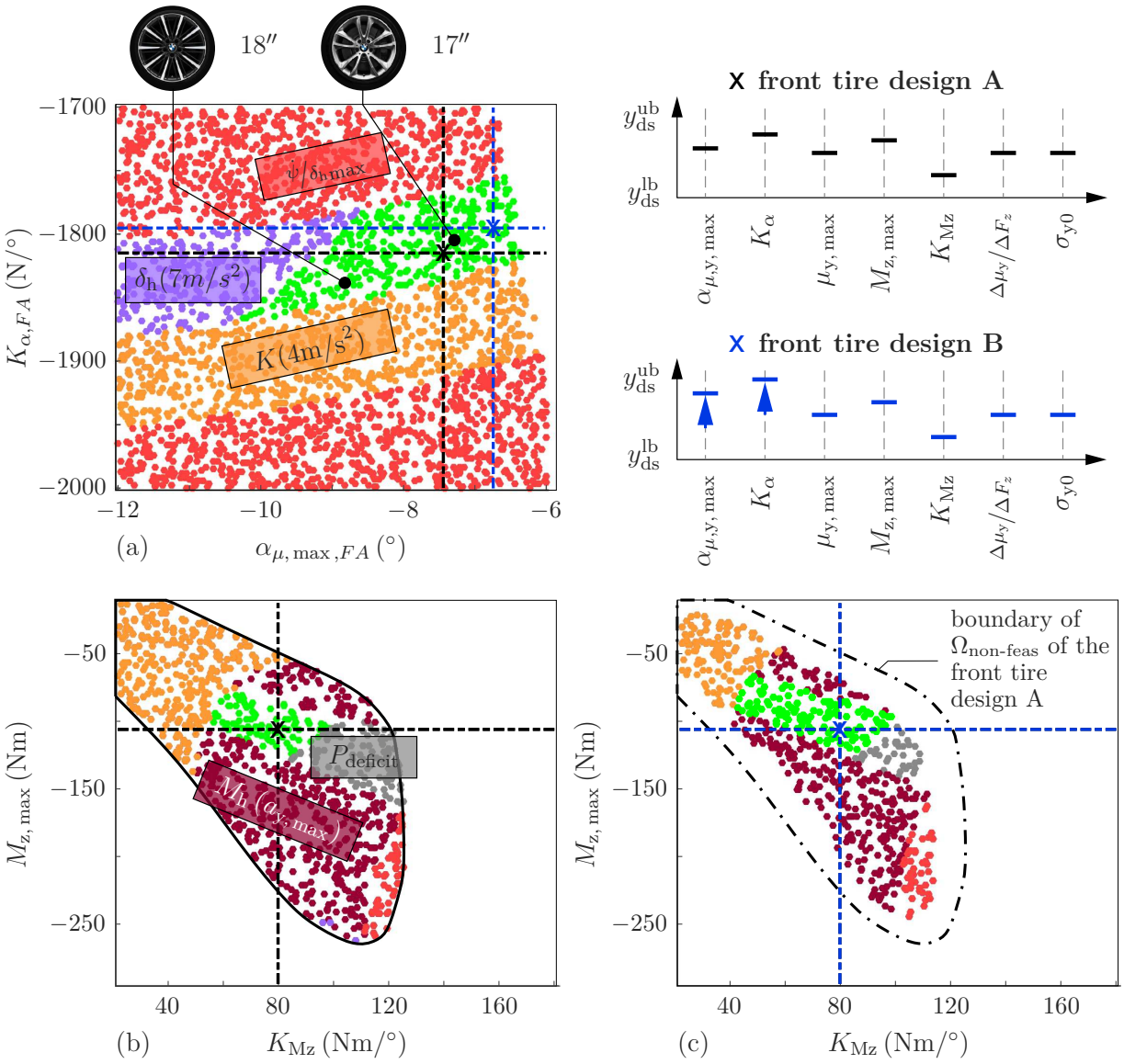


Figure 3.11: The impact of design variables on the solution space and feasible space.

Figure 3.11 shows sections through the design space of FTCs that take feasibility restrictions into account. The non-feasible space is clearly visible because of the white region. Since all designs belonging to the non-feasible space are not taken into account for the solution space calculation, they do not affect the later position of the solution box in an active manner. Therefore, on one hand the size of the solution box is not restricted by those areas as if they were marked as bad designs, but on the other hand, the solution box is also not forced to shift through those areas as if they were good designs. In addition, it is important to note that not all tires belonging to the feasible space are necessarily feasible, but tires belonging to the non-feasible space are definitely not feasible. For visualizing the impact of a particular design point on the surrounding solution space and feasible space, two designs for a front tire are presented in Figure 3.11: **A** and **B**. The difference between both is that **B** has lower cornering stiffness as well as a lower slip angle at the maximum grip in absolute values compared to **B**, see Figure 3.11 (a). Therefore, the associated section view of $M_{\mathrm{z,\,max}}$ vs. K_{Mz} is presented in Figure 3.11 (b) for tire **A** and in Figure 3.11 (c) for tire **B**. It may be observed that the solution space in the section view of tire **B** is bigger, however the feasible space gets smaller. This indicates the importance of considering feasibility in the early development stage, because without it may happen that a large solution space is identified, while only a small subset of the solution space is feasible.

In a second step, the feasibility of tires within the feasible space is evaluated, therefore FTCs from already existing tires may be used. An overlay of the design space and the existing tire portfolio gives an idea as to where tires already exist. In Figure 3.11 (a), a particular tire for 17″ and for 18″ rims is shown. By plotting existing tires into the graph, the solution box may be shifted in order to use existing tires. Depending on the expected leap in development, the prospective tire can be more or less farther away from already existing tires. Figure 3.12 presents the projection of a solution box. The closer to the boundary of $\Omega_{\text{non-feas}}$, the more sparse the sampling. Therefore, regions of good tires that are difficult to realize may be identified early in the development process. The underlying reason for the blurred boundaries of the feasible space and solution space are the projected permissible intervals as described in Subsection 2.3.5. The data of existing tires may also be used to identify ranges of tire detail variables, such as tire width or diameter, for which the system requirements may be satisfied and the resulting sets of FTCs may be realized as well, see (Wimmler et al. 2015). Therefore, trade-offs can be made on evidence-based discussions, e.g., in order to realize a particular set of FTCs, the tire must have a minimum width. Although, this data-based approach gives a good idea about which sets of FTCs are feasible, it is always important to stay in close contact with the supplier and not to be too conservative in terms of requirements, since the requirements on vehicles in the past may differ from the prospective ones.

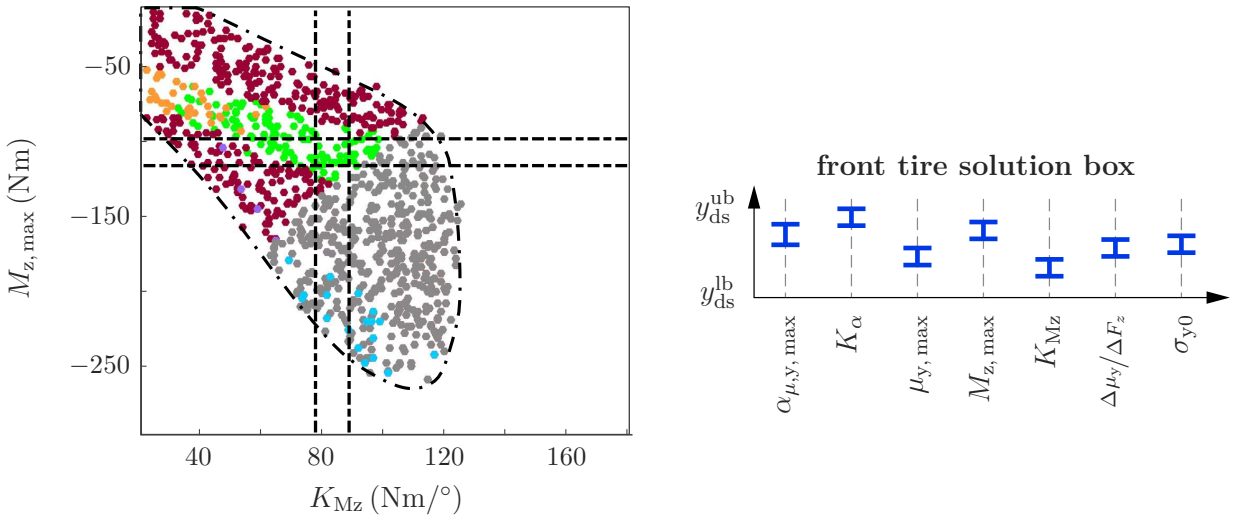


Figure 3.12: Solution spaces for FTCs taking feasibility restrictions into account.

CHAPTER 4

INTEGRATED SUSPENSION DESIGN USING SOLUTION SPACES AND TARGET CASCADING

The integrated design of suspension systems is accomplished by a combination of solution spaces and target cascading. Solution Spaces are derived on functional suspension parameters on the subsystem level (y-level) based on overall vehicle targets (z-level). While deriving those solution spaces, the interaction among the vehicle behavior and all other subsystems may be treated. Furthermore, solution spaces provide the subsystem engineer with greater flexibility in finding a particular first draft design, while modifications up to the final design are still possible.

A great challenge while deriving permissible intervals for the suspension system is to ensure their feasibility. The feasibility of a solution box is guaranteed if at least one design on the x-level exists satisfying all requirements on the y-level.

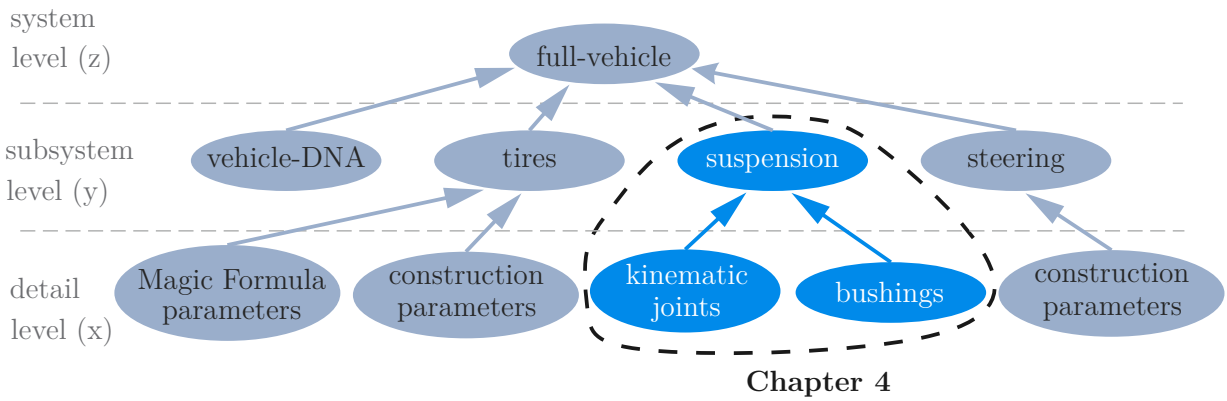


Figure 4.1: The role of the suspension within the interlinked design procedure.

When feasibility of the subsystem requirements is not taken into account, time expensive iteration loops and unused potential are the result. In that case, requirements are derived

on the suspension system and forwarded to the department that designs the suspension geometry. Since the forwarded requirements are not feasible at all, they state their biggest issues and ask for modified requirements. As a result, certain iteration loops go by until finally a consistent design is identified. Nevertheless, an even better solution in terms of performance and robustness would be possible as the product design on a detail level strongly depends on the permitted subsystem performance.

In order to avoid such time expensive iteration loops, the *Integrated Design of Suspension Systems* presented in this chapter uses the information provided by a feasibility optimization between subsystem and detail level in order to find a feasible solution box with a maximum size measure μ. Therefore, the subsystem performance is tuned regarding the requirements assigned to it. However the requirements on the subsystem are tuned regarding the feasible subsystem performance as well.

4.1 Feasibility Optimization

Calculating box-shaped solution spaces is not an appropriate method should many cross-links between design parameters and design goals exist. On one hand, the size of the solution box will suffer and on the other hand, if the box size is increased due to a rotated solution box, the benefit of decoupling the parameter intervals is gone. Instead of calculating solution spaces, a point-based multi-objective optimization is used in order to prove the feasibility of the desired requirements on the suspension system. This optimization is called *feasibility optimization*. It determines a set of feasible design parameters for a system such that the resulting system performance lies close to the center of the forwarded solution box.

Feasibility space. For the feasibility of the permissible intervals, the intervals must overlap the feasibility space. The feasibility space is measured in the output space of the system or subsystem and contains all designs that may be realized by any combination of design parameters. Therefore, different system types of the same group may have different feasibility spaces, e.g., a McPherson and a double wishbone suspension system. Both of them belong to the group of suspension systems, but due to the different type (basic architecture such as the arrangement of the control links differs), the feasibility space for certain performance measures, e.g., change in roll center height RC_{h}, is different as well.

Statement 5 (feasibility of requirements)
The requirements on a system/subsystem are feasible if any combination of design parameters satisfies those requirements.

If the design of the system must satisfy constraints on the input parameters, then the feasibility space shrinks since some designs are excluded from the search for a solution. Therefore, the *constrained feasibility space* must overlap the permissible intervals. In the case of the suspension system, a distance between two kinematic joint positions might be restricted due to geometrical boundary conditions.

In the application example presented in Chapter 6, the feasibility optimization builds the bridge between the functional and geometrical properties of the suspension system. While solution spaces are used to calculate decoupled permissible intervals that may be forwarded to different departments, such as suspension-, tire design department, a feasibility optimization ensures the existence of a set of kinematic joints that satisfies the requirements on the suspension system.

Statement 6 (feasibility space)
The feasibility space includes all designs that may be realized by any combination of design parameters. From a subsystem point of view it is measured in the output space.

Although the feasibility of the requirements is an important issue for all subsystems, the methodology for their evaluation may differ between the different subsystems such as tire and suspension.

In the method presented in Chapter 5, it is taken into consideration that the requirements for the subsystem are derived from the system above as box-shaped solution spaces. Since the solution space of the upper system is restricted from all sides, it is not a so-called one-sided solution space, and therefore all requirements on the subsystem beneath contain an upper and lower bound.

In Section 4.2, the underlying idea of the feasibility optimization is introduced. Therefore, an optimization algorithm that finds a design with a system performance close to the center of the required performance intervals is proposed. As a result, a concrete suspension geometry satisfying all requirements is identified. When developing series-production vehicles, it is desirable that different cars share components for increased profitability. Due to economic analysis, which take development, material and production costs as well as take-rates into account, the identical parts that would be economically viable often are already known. In Sections 4.3 and 4.4, the feasibility optimization is extended in order to consider two different use cases associated with commonality.

4.2 General Problem Statement of the Feasibility Optimization

The first goal of the feasibility optimization is to find a feasible design that satisfies all requirements, and the second goal is to increase robustness against unintended parameter deviations.

4.2.1 Definitions

Design points or designs may be represented as a vector on different abstraction levels ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$). In this work, for the feasibility optimization of a suspension, the design variables belong to the detail level (x-level). Therefore, the design parameters are represented by $\boldsymbol{x} =$

$(x_1, x_2, \ldots, x_m)$, where m is the number of dimensions. As shown in Figure 4.1, the design variables of a suspension system are the kinematic joint positions $K = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}, \boldsymbol{i}\}$ and the stiffnesses of the bushings. The topology of a double wishbone suspension is presented in Figure 4.2. All possible designs within the input space define the design space Ω_{ds}. A single subsystem performance criterion is calculated by

$$y = \mathrm{f}_2(\boldsymbol{x}) \tag{4.1}$$

with the performance function f_2. The complete subsystem performance is represented by $\boldsymbol{y} = (y_1, y_2, \ldots, y_t)$, where t is the number of targets on the regarded subsystem. For the complex suspension model used in this work, an analytical performance function is unknown. Therefore, numerical optimization is used in order to find the optimal solution. In the approach presented here, the performance of the subsystem should lie in the center of the required intervals in order to increase the robustness of the design. The center of a permissible interval is calculated by

$$y_j^{\mathrm{target}} = \frac{y_j^{\mathrm{ub}} + y_j^{\mathrm{lb}}}{2} \tag{4.2}$$

with the lower threshold value y^{lb} and the upper one y^{ub} of the associated interval. In order to reduce the distance between the center of the required intervals and the suspension's performance, the design variables are adapted by numerical optimization.

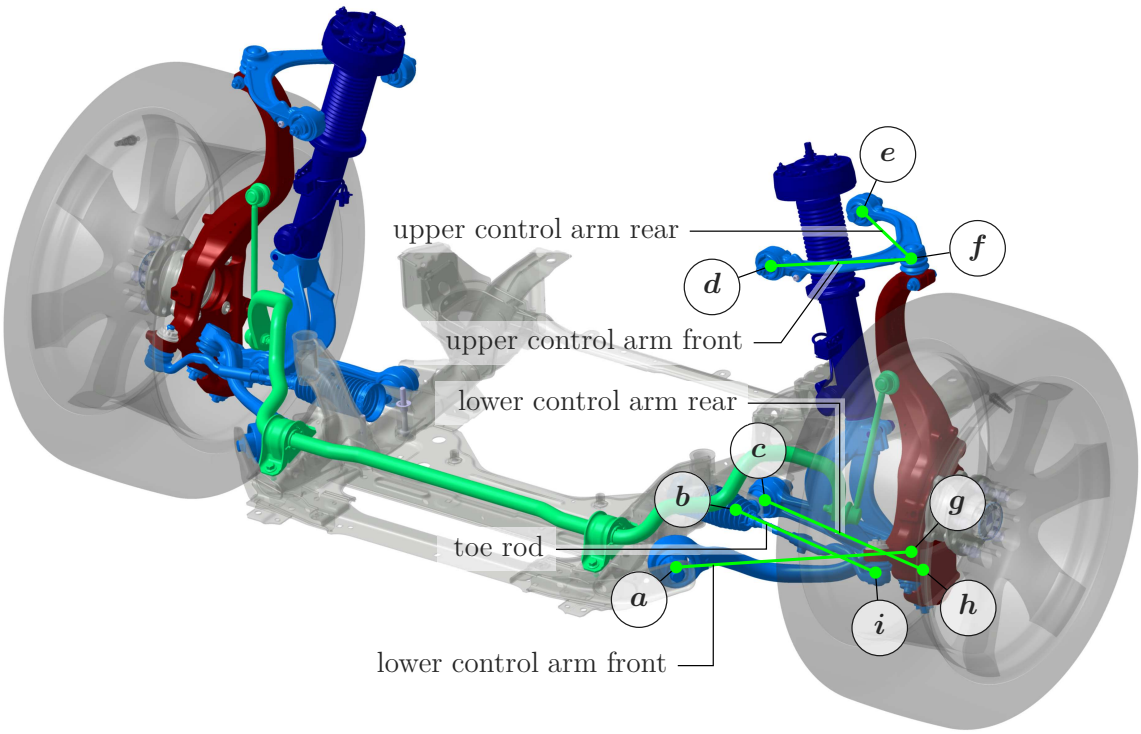


Figure 4.2: Topology of the double wishbone suspension optimized in this work.

4.2.2 Problem Statement for Finding a Feasible and Robust Design

For t design targets, the distance of the performance value y_j and the center of the associated required interval y_j^{target} should be minimized. For weighting reasons between different subsystem targets, the distance between a particular subsystem performance y_j and the center of the required interval y_j^{target} is measured in relation to the associated permissible interval $\boldsymbol{I}_j = (y_j^{\text{lb}}, y_j^{\text{ub}})$. In engineering design problems, often it is important to find any feasible solution that satisfies all the requirements before optimizing the robustness. Therefore, a penalty factor $p(y_j)$ is used, if a performance value does not meet the requirements such that $y_j < y_j^{\text{lb}}$ or $y_j > y_j^{\text{ub}}$. Consequently, if no design satisfying all requirements could be found, the feasibility optimization will find a design as close as possible according to the performance function shown in Equation (4.3). As a result, the number of requested requirement intervals that must be rebalanced on the level above is reduced.

$$
\begin{aligned}
&\underset{x_1,x_2,\ldots,x_m}{\text{minimize}} \quad \varphi_{\text{feas}}\left(\boldsymbol{y}(\boldsymbol{x})\right) \\
&\text{with} \qquad \varphi_{\text{feas}} = -0.5\, d \sum_{j=1}^{t} \left| \frac{y_j - y_j{}^{\text{target}}}{y_j^{\text{ub}} - y_j^{\text{lb}}} \right| + p\, \frac{\min(|y_j - y_j^{\text{lb}}|, |y_j^{\text{ub}} - y_j|)}{y_j^{\text{ub}} - y_j^{\text{lb}}} \\
&\text{and} \qquad p = \begin{cases} 0 & \text{if } y_j^{\text{lb}} < y_j < y_j^{\text{ub}} \\ 1 & \text{if } y_j < y_j^{\text{lb}} \text{ or } y_j > y_j^{\text{ub}} \end{cases}
\end{aligned}
\tag{4.3}
$$

4.3 Design of an Axle Subframe for a Vehicle Platform

If an axle subframe is developed for a new vehicle platform, it is desirable to reduce the number of variants. Therefore, the feasibility optimization presented in Section 4.2 can be extended to handle multiple suspension systems that are intended to share the same axle subframe.

4.3.1 Definitions

For the consideration of multiple suspension systems, additional denotations of the variables are introduced. Since the requirements on the system performance $\boldsymbol{y}^{\text{lb}}$ and $\boldsymbol{y}^{\text{ub}}$, the system performance $\boldsymbol{y}$ itself and some of the design variables $\boldsymbol{x}$ are allowed to differ between the systems they must be considered separately. Therefore, they are denoted by $\text{s}1, \text{s}2, \ldots, \text{s}n$ in the left lower corner for the associated system, where n is the number of subsystems that should use the same component. Since the optimization is performed on a surrogate model which is based on the relative displacements of kinematic joint positions related to a reference model, their coordinates must be transformed into an absolute coordinate system in order to allow surrogate models that are based on different reference models. If "v" shows up in the left upper corner of a design variable, it is measured within the construction coordinate system of the vehicle. According to (Haken 2015), the construction coordinate system is right-handed with its origin on a line between the two front

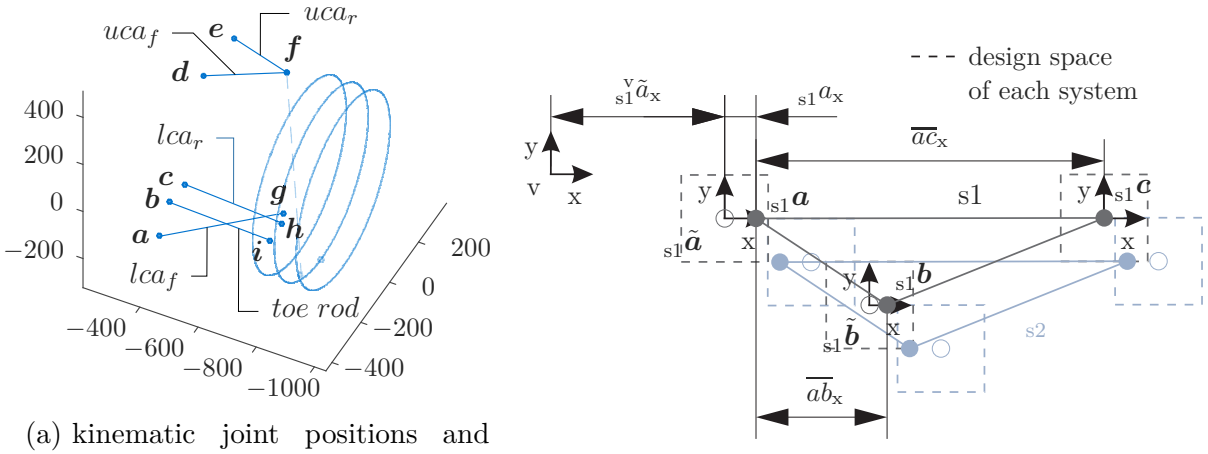
wheel centers in the middle of the vehicle. The positive x-axis points towards the rear of the vehicle, while the positive z-axis points upwards, perpendicular to the ground plane. If no index exists in the left upper corner, the displacement related to the absolute coordinates of the reference model is presented. The index in the right upper corner indicates whether the upper or lower bound of the design space is shown, while the one in the right lower corner refers to the kinematic joint and direction. The general indexing is stated in Equation (4.4).

$$\overset{\text{coordinate system}}{\underset{\text{system}}{}}\mathit{kinematic\ joint}\overset{\text{lower bound or upper bound of the design space}}{\underset{\text{kinematic joint, direction}}{}} \tag{4.4}$$

After the requirements on all suspension systems of the vehicles to be developed are defined, the kinematic joints that belong to the axle subframe need to be identified. In the case of a double wishbone suspension, the kinematic joints $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, see Figure 4.3 (a), belong to the axle subframe. Since the width of the subframe may neither shrink nor extend, their y-position must be the same for each vehicle (measured in the absolute coordinate system). On the other hand, the associated x-/ and z-positions are allowed to differ between each suspension system by an offset value, as the subframe may be installed in other positions. Nevertheless, the relative distances between those kinematic joint positions must remain the same in order to use the same component. Therefore, a different parameterization is selected for the optimization of an axle subframe for a vehicle platform. The position of the independent kinematic joints $I = \{\boldsymbol{d},\boldsymbol{e},\boldsymbol{f},\boldsymbol{g},\boldsymbol{h},\boldsymbol{i}\}$ is parameterized as described in Section 4.2. Their positions may be denoted by ${}_{si}\boldsymbol{x}_{\text{ind}}$ for a particular system i. For the position of the common kinematic joints $C = \{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\}$, the absolute distance $\overline{\boldsymbol{aj}}$ between a reference joint $\boldsymbol{a}$ and an arbitrary kinematic joint $\boldsymbol{j}$ that belongs to the subframe is used to describe the geometry of the subframe. The absolute distance in each direction is given by $\overline{\boldsymbol{aj}} = \left(\overline{aj}_{\text{x}}, \overline{aj}_{\text{y}}, \overline{aj}_{\text{z}}\right)^{\intercal}, \boldsymbol{j} \in C$ and may be calculated according to Equation (4.5).

$$\overline{\boldsymbol{aj}} = {}^{\text{v}}_{si}\tilde{\boldsymbol{j}} - {}^{\text{v}}_{si}\tilde{\boldsymbol{a}},\ \boldsymbol{j} \in C \tag{4.5}$$

Additionally, the offset vector $\boldsymbol{a}_d = ({}_{\text{s1}}a_d, {}_{\text{s2}}a_d, ..., {}_{sn}a_d)$ defines the position of the subframe within the absolute coordinate system for each vehicle, where d denotes whether the x-/ or z-direction is meant. Therefore, miscellaneous mounting positions of the subframe are enabled. The offset values of $\boldsymbol{a}_{\text{x}}$ and $\boldsymbol{a}_{\text{z}}$ refer to the reference model of the associated suspension system such that the position of the reference model ${}^{\text{v}}_{si}\tilde{\boldsymbol{a}}$ must be added to ${}_{si}\boldsymbol{a}$ in order to obtain the optimized position within the absolute coordinate system. For a two-dimensional example of the parameterization see Figure 4.3 (b). In order to realize the same width of the subframe within all vehicles, one scalar offset value in the y-direction is used for all vehicles with the same subframe. The offset in y-direction refers to the first system s1 such that $a_{\text{y}} = {}_{\text{s1}}a_{\text{y}}$. Therefore, the absolute value of $\boldsymbol{a}$ in y-direction for all vehicles is calculated by ${}^{\text{v}}a_{\text{y}} = a_{\text{y}} + {}^{\text{v}}_{\text{s1}}\tilde{a}_{\text{y}}$. Since each kinematic model uses relative displacements as input, the y-position is transformed in the associated relative coordinate system by ${}_{si}a_{\text{y}} = {}^{\text{v}}a_{\text{y}} - {}^{\text{v}}_{si}\tilde{a}_{\text{y}}$.



(a) kinematic joint positions and wireframe of a double wishbone suspension system

(b) on the example of the front axle subframe

Figure 4.3: Overview of the kinematic joint position parameterization.

The design space borderlines of the absolute distances are calculated by the Equations (4.6) to (4.9) in order to find the region covered by all kinematic models. Therefore, first, the extreme values ${}_{si}\overline{aj}_d^{\text{ub ds}}$ and ${}_{si}\overline{aj}_d^{\text{lb ds}}$ between the associated kinematic joints are identified separately for each system i and dimension d by Equation (4.6), (4.7), and second, according to Equation (4.8), the minimum of all upper bounds is used as upper bound $\overline{aj}_d^{\text{ub ds}}$ for the commonality optimization, while the maximum of the lower bounds gives the lower bound $\overline{aj}_d^{\text{lb ds}}$, see Equation (4.9).

$$ {}_{si}\overline{aj}_d^{\text{ub ds}} = \max\left({}^{\text{v}}_{si}j_d^{\text{ub ds}} - {}^{\text{v}}_{si}a_d^{\text{lb ds}}, {}^{\text{v}}_{si}j_d^{\text{lb ds}} - {}^{\text{v}}_{si}a_d^{\text{ub ds}}\right), j \in C \tag{4.6} $$

$$ {}_{si}\overline{aj}_d^{\text{lb ds}} = \min\left({}^{\text{v}}_{si}j_d^{\text{ub ds}} - {}^{\text{v}}_{si}a_d^{\text{lb ds}}, {}^{\text{v}}_{si}j_d^{\text{lb ds}} - {}^{\text{v}}_{si}a_d^{\text{ub ds}}\right), j \in C \tag{4.7} $$

$$ \overline{aj}_d^{\text{ub ds}} = \min\left({}_{\text{s1}}\overline{aj}_d^{\text{ub ds}}, {}_{\text{s2}}\overline{aj}_d^{\text{ub ds}}, \ldots, {}_{sn}\overline{aj}_d^{\text{ub ds}}\right), j \in C \tag{4.8} $$

$$ \overline{aj}_d^{\text{lb ds}} = \max\left({}_{\text{s1}}\overline{aj}_d^{\text{lb ds}}, {}_{\text{s2}}\overline{aj}_d^{\text{lb ds}}, \ldots, {}_{sn}x_d^{\text{lb ds}}\right), j \in C \tag{4.9} $$

The optimization vector $\boldsymbol{x}_0$, which follows the proposed parameterization, consists of: first, the distances between common kinematic joints $\overline{\boldsymbol{aj}}$, second, the offset values ${}_{\text{s1}}a_{\text{x}} \ldots {}_{sn}a_{\text{z}}$, and third, the positions of the independent kinematic joints for each system ${}_{si}\boldsymbol{x}_{\text{ind}}$. Figure 4.4 shows the entire procedure for the assignment between the optimization variables and design variables of the suspension systems.

data: optimization vector $\boldsymbol{x_0} = \left(\overline{\boldsymbol{aa}}, \overline{\boldsymbol{ab}}, ..., \overline{\boldsymbol{aj}},\ \boldsymbol{a}_{\mathrm{x}}, a_{\mathrm{y}}, \boldsymbol{a}_{\mathrm{z}},\ {}_{\mathrm{s}1}\boldsymbol{x}_{\mathrm{ind}}, {}_{\mathrm{s}2}\boldsymbol{x}_{\mathrm{ind}}, ..., {}_{sn}\boldsymbol{x}_{\mathrm{ind}}\right)$
for i = 1,2,..., number of systems
  generating the shifting vector:
  ${}_{si}\boldsymbol{a} = (\boldsymbol{a}_{\mathrm{x}}(i), a_{\mathrm{y}}, \boldsymbol{a}_{\mathrm{z}}(i))$
  original position of the common subframe:
  ${}^{\mathrm{v}}_{si}\tilde{\boldsymbol{a}} = ({}^{\mathrm{v}}\tilde{\boldsymbol{a}}_{\mathrm{x}}(i), {}^{\mathrm{v}}\tilde{a}_{\mathrm{y}}, {}^{\mathrm{v}}\tilde{\boldsymbol{a}}_{\mathrm{z}}(i))$
  calculating the optimized position of the common subframe:
  ${}^{\mathrm{v}}_{si}\boldsymbol{a} = {}_{si}\boldsymbol{a} + {}^{\mathrm{v}}_{si}\tilde{\boldsymbol{a}}$
  for all common kinematic joints $\boldsymbol{j} \in K$
    calculation of the coordinates in the coordinate system of the vehicle:
    ${}^{\mathrm{v}}_{si}\boldsymbol{j} = \overline{\boldsymbol{aj}} + {}^{\mathrm{v}}_{si}\boldsymbol{a}$
    calculation of the coordinates in the coordinate system of the element:
    ${}_{si}\boldsymbol{j} = {}^{\mathrm{v}}_{si}\boldsymbol{j} - {}^{\mathrm{v}}_{si}\tilde{\boldsymbol{j}}$
  end
  amalgamation of common and independent kinematic joint positions $K = C \cup I$ in order to obtain the input vector ${}_{si}\boldsymbol{x}$ of the current system i
  calculate subsystem performance:
  ${}_{si}\boldsymbol{y}$=$\mathrm{f}_2({}_{si}\boldsymbol{x})$
end
result: performance measure of each subsystem

Figure 4.4: Algorithm for the assignment between parameters of the optimization vector and design parameters of the simulation models.

4.3.2 Problem Statement for Platform Design

The resulting vectors ${}_{si}\boldsymbol{x}$ deliver the input for the simulation models that calculate the performance of the subsystems ${}_{si}\boldsymbol{y}$. Based on the performance function shown in Equation (4.10), the subsystem performance of each suspension system is evaluated and totaled.

$$
\begin{aligned}
&\underset{{}_{\mathrm{s}1}\boldsymbol{x}, {}_{\mathrm{s}2}\boldsymbol{x}, ..., {}_{sn}\boldsymbol{x}}{\text{minimize}} && \varphi_{\mathrm{feas}}\left({}_{\mathrm{s}1}\boldsymbol{y}({}_{\mathrm{s}1}\boldsymbol{x}), {}_{\mathrm{s}2}\boldsymbol{y}({}_{\mathrm{s}2}\boldsymbol{x}), ..., {}_{sn}\boldsymbol{y}({}_{sn}\boldsymbol{x})\right) \\
&\text{with} && \varphi_{\mathrm{feas}} = {}_{\mathrm{s}1}\varphi_{\mathrm{feas}} + {}_{\mathrm{s}2}\varphi_{\mathrm{feas}} + ... + {}_{sn}\varphi_{\mathrm{feas}} \\
&\text{and} && {}_{si}\varphi_{\mathrm{feas}} = -0.5\, t \sum_{j=1}^{t} \left| \frac{{}_{si}y_j - {}_{si}y_j{}^{\mathrm{target}}}{{}_{si}y_j^{\mathrm{ub}} - {}_{si}y_j^{\mathrm{lb}}} \right| + p\, \frac{\min(\left|{}_{si}y_j - {}_{si}y_j^{\mathrm{lb}}\right|, \left|{}_{si}y_j^{\mathrm{ub}} - {}_{si}y_j\right|)}{{}_{si}y_j^{\mathrm{ub}} - {}_{si}y_j^{\mathrm{lb}}} \qquad (4.10) \\
&\text{and} && p = \begin{cases} 0 & \text{if } {}_{si}y_j^{\mathrm{lb}} < {}_{si}y_j < {}_{si}y_j^{\mathrm{ub}} \\ 1 & \text{if } {}_{si}y_j < {}_{si}y_j^{\mathrm{lb}} \text{ or } {}_{si}y_j > {}_{si}y_j^{\mathrm{ub}} \end{cases}
\end{aligned}
$$

4.4 Architecture Design of Control Links for a Vehicle Platform

In the case of different vehicles that should share one or more control links, the feasibility optimization presented in Section 4.2 is extended to handle multiple systems and constraints. Due to the different types of control links that exist and the required flexibility in selecting common control links, a re-parameterization as proposed in Section 4.3 would quickly become very complex. Therefore, non-linear constraints are used in order to optimize control links for commonality. Since no re-parameterization is necessary, creating the optimization vector $\boldsymbol{x}_0$ it is more straightforward compared with the subframe optimization in Section 4.3 such that the sequence of the relative kinematic joint displacements of each subsystem result in the optimization vector $\boldsymbol{x}_0 = ({}_{s1}\boldsymbol{x}, {}_{s2}\boldsymbol{x}, ..., {}_{sn}\boldsymbol{x})$. The performance function, which is applied to optimize the subsystem performance, is the same as described in Equation (4.10).

Depending on the topology of a particular control link, the number of constraints varies. A push/pull rod, for example, requires only one constraint for its length in order to guarantee the same geometry, while an A-arm requires at least three constraints for the distances between the associated kinematic joints in order to be fully defined. A generic way to set the constraints independently of the control link topology is shown in Figure 4.5. The control links topology is identified by the kinematic joints which belong to the connection line that should be equal for various systems. For example, the development engineer can decide whether he wants to have a common uca_f, uca_r, lca_f, lca_r, *toe rod* or any combination of them. If the A-arm of the suspension shown in Figure 4.3 should be the same, uca_f and uca_r must be selected. Accordingly, the algorithm recognizes that uca_f and uca_r belong to the same part, since both connecting lines $\overline{\boldsymbol{df}}$ and $\overline{\boldsymbol{ef}}$ are attached to the same kinematic joint $\boldsymbol{f}$. For that reason, both connection lines belong to the same part such that the third nonlinear constraint ${}_{s1}\overline{\boldsymbol{de}} = {}_{s2}\overline{\boldsymbol{de}}$ may be created automatically. All distances between kinematic joints that are connected with a constraint are stored in $\boldsymbol{C}$

data: optimization vector $\boldsymbol{x_0}$, constraint matrix $\boldsymbol{C} = ({}^{v}\boldsymbol{d}, {}^{v}\boldsymbol{f}; {}^{v}\boldsymbol{e}, {}^{v}\boldsymbol{f}; {}^{v}\boldsymbol{d}, {}^{v}\boldsymbol{e})$
for p = 1,2,..., number of nonlinear constraints for each additional system
  $\boldsymbol{j} = \boldsymbol{C}(p,1)$, $\boldsymbol{k} = \boldsymbol{C}(p,2)$
  for i = 1,2,..., number of systems−1
    definition of the nonlinear equality constraints
    $0 = |{}_{si}\overrightarrow{\boldsymbol{jk}}| - |{}_{si+1}\overrightarrow{\boldsymbol{jk}}|$, $\boldsymbol{j} \cup \boldsymbol{k} \in \boldsymbol{C}$
  end
  distance between the current connection line must be the same in all systems
  $|{}_{s1}\overrightarrow{\boldsymbol{jk}}| = |{}_{s2}\overrightarrow{\boldsymbol{jk}}| = ... = |{}_{sn}\overrightarrow{\boldsymbol{jk}}|$
end
result: $p(i-1)$ nonlinear equality constraints

Figure 4.5: Algorithm for generic creation of nonlinear equality constraints.

CHAPTER 5

TARGET CASCADING FOR VEHICLES AND PLATFORMS USING SOLUTION SPACES

To save development time and to meet all specified top-level design goals, it is essential to ensure that all influencing module properties interact correspondingly, while the feasibility of the solution space is ensured as well. The novel method proposed in this chapter consists of three tools that are necessary in order to derive requirements top-down across the introduced three abstraction levels. Each tool is an individual optimization algorithm which is applied sequentially in three stages. One tool is the feasibility optimization which was presented in Chapter 4. It ensures the feasibility of the permissible intervals associated with the suspension system. Feasibility of the requirements on the tire may be considered using the method proposed in Chapter 3. The other two tools are an extension of the robust optimization and solution box optimization presented in Subsection 2.3.3 and Subsection 2.3.4. In Section 5.2 and 5.3, two different algorithms for the computation of feasible solution spaces by a combination of the three tools are presented. Therefore, a decision-based, semi-automatic algorithm, as well as a fully automated algorithm, are introduced.

5.1 Overall design strategy

As mentioned in Section 1.5, the vehicle is clustered hierarchically into three abstraction levels. The different abstraction levels distinguish between requirements on the system, subsystem and detail level properties of the vehicle. Therefore, according to the associated abstraction level, the properties are denoted by the vectors $\boldsymbol{z}$, $\boldsymbol{y}$, and $\boldsymbol{x}$. The system performance $\boldsymbol{z}$ is computed differently depending on the quantitative simulation model used. Since the overall vehicle performance $\boldsymbol{z}$ may be calculated as a function of $\boldsymbol{y}$, which in turn is a function of $\boldsymbol{x}$, $\boldsymbol{z}$ may be computed by $\boldsymbol{z} = \mathrm{f}_1(\boldsymbol{y}(\boldsymbol{x}))$ as well as $\boldsymbol{z} = \mathrm{f}_3(\boldsymbol{x})$.

An overview of the overall design strategy is given in Figure 5.1. The parameters belonging to the system level $\boldsymbol{z}$ are also called CVs. For those objective criteria, permissible

intervals are specified by an engineer based on the expectations of the customer and objectification knowledge. The resulting region is a box-shaped space on the z-level which comprises all combinations of the vehicle performance measures that satisfy the customer expectations. Therefore, it may be called a target region or a solution box on the z-level $\Omega^{\mathrm{z}}_{\mathrm{sb}}$, although it is within the output space. Nevertheless, if an additional subjective level would be considered above, the system level would be the associated input space. Based on the target region $\Omega^{\mathrm{z}}_{\mathrm{sb}}$ and the causal relations $\boldsymbol{z} = \mathrm{f}_1(\boldsymbol{y})$, a solution space $\Omega^{\mathrm{y}}_{\mathrm{ss}}$ is identified on the y-level. The *solution space* is a subset of the design space $\Omega^{\mathrm{y}}_{\mathrm{ss}} \subset \Omega^{\mathrm{y}}_{\mathrm{ds}}$ and contains all designs on the y-level that satisfy the overall performance targets, see Equation (5.1).

$$\Omega^{\mathrm{y}}_{\mathrm{ss}} = \{\boldsymbol{y} \in \Omega^{\mathrm{y}}_{\mathrm{ds}} \,|\, \boldsymbol{z}^{\mathrm{lb}} < \mathrm{f}_1(\boldsymbol{y}) < \boldsymbol{z}^{\mathrm{ub}}\} \tag{5.1}$$

In order to realize a vehicle with the desired top-level properties, the requirements derived on the subsystem level properties must be feasible by adjusting the detail level properties. The region on the y-level, which can be assigned to design sets of a particular suspension type on the x-level, is called feasible space. Therefore, the *feasible space* is defined by all designs on the y-level that can be associated with a design on the x-level.

$$\Omega^{\mathrm{y}}_{\mathrm{feas,A}} = \{\, \boldsymbol{y} \in \Omega^{\mathrm{y}}_{\mathrm{ds}} \,|\, \exists\, \boldsymbol{x} : \boldsymbol{y} = \mathrm{f}_2(\boldsymbol{x})\} \tag{5.2}$$

The region of the feasible space depends on the design space of the particular subsystem $\Omega^{\mathrm{x}}_{\mathrm{ds,A}}$ as well as the causal relations between detail level and subsystem level $\mathrm{f}_2(\boldsymbol{x})$. Therefore, according to Equation (5.2) the feasible space might be different depending on the subsystem type used, e.g., suspension system A results in a different feasibility space than suspension system B. If an overlap between solution space and feasible space exists on the y-level, the requirements on the subsystem are feasible. This overlapping region is called *feasible solution space* $\Omega^{\mathrm{y}}_{\mathrm{fss,A}}$. Therefore, according to Equation (5.3) all designs that belong to the solution space, as well as to the feasible space, are included.

$$\Omega^{\mathrm{y}}_{\mathrm{fss,A}} = \Omega^{\mathrm{y}}_{\mathrm{ss}} \cap \Omega^{\mathrm{y}}_{\mathrm{feas,A}} = \{\boldsymbol{y} \,|\, \boldsymbol{y} \in \Omega^{\mathrm{y}}_{\mathrm{ss}} \,\wedge\, \boldsymbol{y} \in \Omega^{\mathrm{y}}_{\mathrm{feas,A}}\} \tag{5.3}$$

If no overlap between solution space and feasible space exists, the design problem is *infeasible*. Therefore, the following possibilities exist in order to establish feasibility:

- Relaxation of the top-level requirements $\Omega^{\mathrm{z}}_{\mathrm{sb}}$.
- Implementation of another, possibly better, and more expensive subsystem type, with different $\Omega^{\mathrm{x}}_{\mathrm{ds,A}}$ or $\mathrm{f}_2(\boldsymbol{x})$, in order to ensure the feasibility of the requirements.

However, since the complete solution space $\Omega^{\mathrm{y}}_{\mathrm{ss}}$ cannot be described for high dimensional design problems, a solution box is used as sub-space of the solution space. The requirements on the subsystem performance are described by the associated permissible intervals of the solution box $\Omega^{\mathrm{y}}_{\mathrm{sb}}$. Therefore, both the solution box and the feasible space need to overlap in order to ensure the feasibility of the requirements on the subsystem. Usually, the biggest solution box is sought in order to realize a robust design in the center of it, increase the flexibility during the development process and provide alternative design scenarios. Within

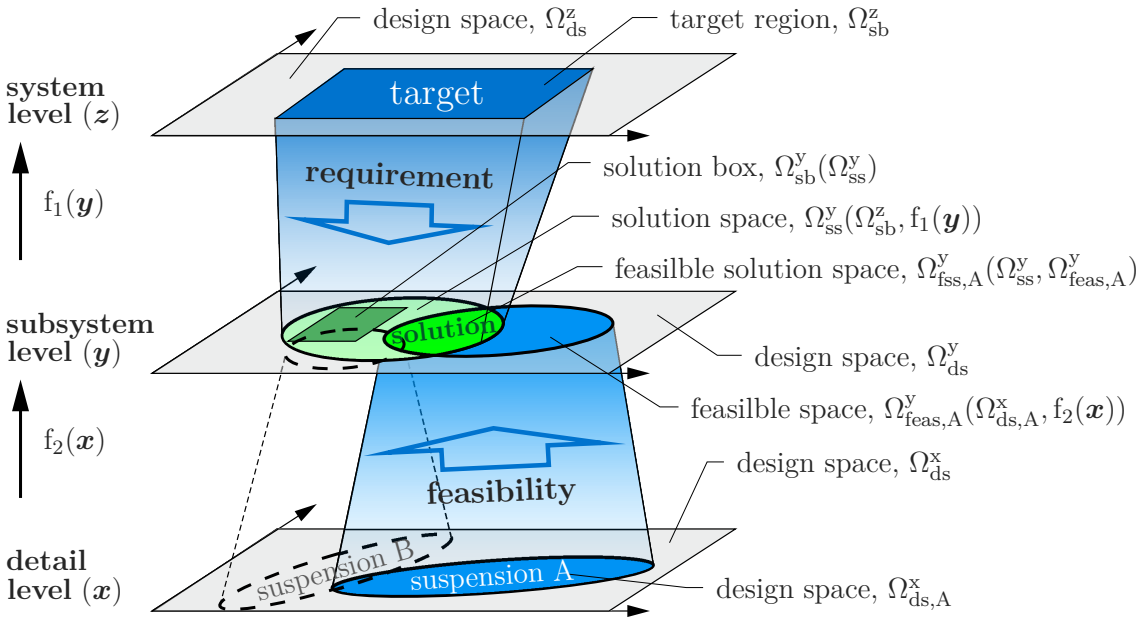


Figure 5.1: General design strategy for Target Cascading.

the computed solution box, all top-level requirements are satisfied. For high dimensional design problems with complex subsystems, e.g., suspension systems, the solution box is significantly smaller than the solution space. Since the solution box does not contain all good designs, it is possible that the derived solution box is not feasible, even if the design problem itself is feasible. In that particular case, the solution box must be shifted towards the feasible space, even if this leads to a reduction of the box size measure μ_{sb}. For complex systems, the feasible space cannot be described analytically. Feasibility optimization can indicate whether the solution box is feasible or not. Therefore, a design on the detail level $\boldsymbol{x}$ is sought such that, according to Chapter 4, the distance between subsystem performance $\boldsymbol{y}$ and the center of the permissible intervals is minimized. As a result, a feasible and robust design should be identified.

In the following two sections, two algorithms are presented which describe how to derive feasible requirements on the subsystem level. The proposed algorithms contain two general steps:

- In a *requirement step*, permissible intervals are derived on subsystem properties in the form of hyperdimensional solution boxes.
- In a *feasibility step*, the feasibility of the former derived requirements is checked and a robust design is proposed.

Therefore, requirements may be derived in a top-down process as permissible intervals, while the consistency between the different abstraction levels of the vehicle is ensured. In a design process along the V-model, this approach leads to transparency, flexibility, and robustness. The proposed methods can be easily extended to design problems with more than three abstraction levels, however, each additional level comes along with a

reduction of the forwarded solution space. Since only a part of the complete solution space is exploited and propagated as target region to the level beneath, it must be evaluated whether an additional abstraction level is useful. Therefore, e.g., the required level of detail, the shape of the solution space as well as the availability of quantitative simulation models and the underlying causal relationships need to be considered. In addition, it must be considered that solution boxes need to be computed on all abstraction levels that have another abstraction level beneath. Hence, the consistency of the vehicle goals is ensured. On the lowest abstraction level, only a feasibility optimization may be performed as the V-model goes up again from then on.

5.2 Automatic Algorithm

A fully automated approach may be preferred when it is difficult to find an overlap between solution box and feasible space. Therefore, a fully automated method for a three-level optimization using solution spaces is presented in this chapter. The method consists of three essential stages which are presented in Figure 5.2. In the initial stage 1, a design point on the y-level is optimized for robustness such that it lies close to the center of the solution space according to Subsection 2.3.3. Afterward, when stage 2 is applied in the first iteration loop, the robust design resulting from stage 1 is expanded to a solution box as described in Subsection 2.3.4. The third stage evaluates whether it is possible to find at least one design on the x-level with a performance satisfying the requirements derived by the solution box on the y-level. Therefore, the feasibility optimization described in Section 4.1 is used.

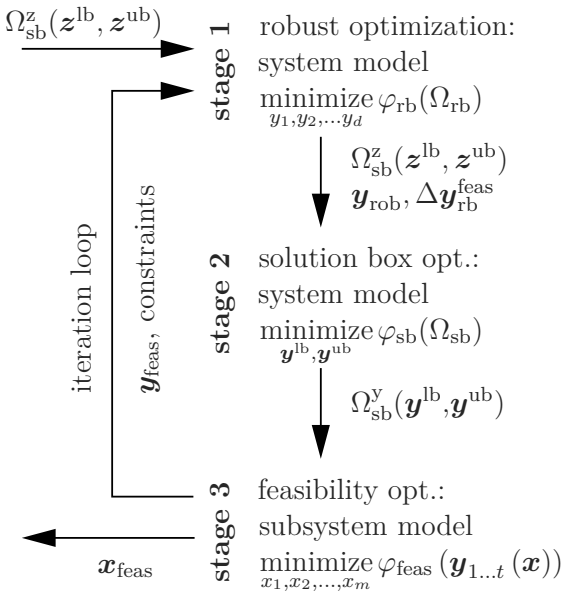


Figure 5.2: Automatic Algorithm – workflow.

If no feasible design that satisfies the subsystem requirements could be identified, or if such a design could be identified but is not robust enough, another iteration loop is necessary. Starting with the second iteration loop, the results of the previous feasibility optimization are considered for the architecture design in stage 1 and stage 2. Therefore, the optimization algorithms of both stages are extended. In Subsection 5.2.1, the extension of the robust design optimization is explained in detail, while the extension of the solution box optimization is presented in Subsection 5.2.2.

5.2.1 Extension of the Robust Optimization for Target Cascading

In order to ensure the feasibility of the derived solution box for complex subsystems, the detail level is considered as well. A solution box is called feasible if at least one design on the detail level $\boldsymbol{x}$ satisfies all requirements set by the solution box $\Omega^{\mathrm{y}}_{\mathrm{sb}}$, see Equation (5.4).

$$\exists\, \boldsymbol{x} : \mathrm{f}_2(\boldsymbol{x}) \in \Omega^{\mathrm{y}}_{\mathrm{sb}} \tag{5.4}$$

At the beginning of stage 2, the distance between the starting point of the solution box optimization $\boldsymbol{y}_{\mathrm{rob}}$ and the closest feasible design $\boldsymbol{y}_{\mathrm{feas}}$ should not be too large. The smaller the distance, the higher the probability that the solution box is able to grow into the feasible solution space $\Omega^{\mathrm{y}}_{\mathrm{fss,A}}$. From a computational expense perspective, using one particular design point to influence the position of the final solution box is better suited compared to the use of solution boxes. Therefore, a robust optimization is applied in stage 1. While multiple design points need to be evaluated in order to evaluate one particular solution box, the robust optimization requires only one evaluation for each design. From the second iteration loop onwards, the position of the closest design which is feasible on the level below $\boldsymbol{y}_{\mathrm{feas}} = \mathrm{f}_2(\boldsymbol{x})$ is considered during the robust optimization. Therefore, the origin of the later solution box is shifted towards the feasible solution space.

Consequently, the performance of a particular robust design is not only measured by the size of its robustness box, as it is described in Subsection 2.3.3, but also by the position of the robustness box. A robustness box around the robust design defines the allowed region of the feasible design. If the feasible design lies outside of the allowed region, a penalty is added to the performance of the robust design. The amount of the penalty depends on the shortest distance between the allowed region and the feasible design. The allowed region around the robust design $\boldsymbol{y}_{\mathrm{rob}}$ is defined by scaling the robustness box Ω_{rb} to Ω^{*}_{rb}. Since not all designs within the robustness box are good regarding the top-level requirements, a scaling factor f_{scale} is used to specify the allowed maximum distance between the feasible and robust design in relation to the boundaries of the robustness box. Therefore, both threshold values of each robustness interval are scaled by the Equations (5.5) and (5.6).

$$y^{\mathrm{lb}*}_{\mathrm{rob},j} = y_{\mathrm{rob},j} - \Delta y^{\mathrm{lb}}_{\mathrm{rob},j}\, f_{\mathrm{scale}} \tag{5.5}$$

$$y^{\mathrm{ub}*}_{\mathrm{rob},j} = y_{\mathrm{rob},j} - \Delta y^{\mathrm{ub}}_{\mathrm{rob},j}\, f_{\mathrm{scale}} \tag{5.6}$$

The smaller the scaling factor, the smaller the distance should be between the center of the robustness box and the feasible design of the previous iteration loop. Subsequently, the allowed region of the feasible design around the robust design is represented by

$$\Omega^{*}_{\mathrm{rb}} = I^{*}_{\mathrm{rob},1} \times I^{*}_{\mathrm{rob},2} \times \cdots \times I^{*}_{\mathrm{rob},d} \tag{5.7}$$

where $I^{*}_{\mathrm{rob},j} = [y^{\mathrm{lb}*}_{\mathrm{rob},j}, y^{\mathrm{ub}*}_{\mathrm{rob},j}]$ represents the j-th robustness interval, with $j = 1,2,\ldots,d$. The underlying idea of defining an allowed region is that $\boldsymbol{y}_{\mathrm{rob}}$ is not only adapted in order to maximize the size measure of the robustness box μ_{rb}, but also to reduce the distance between the robustness box and the feasible design $\boldsymbol{y}_{\mathrm{feas}}$.

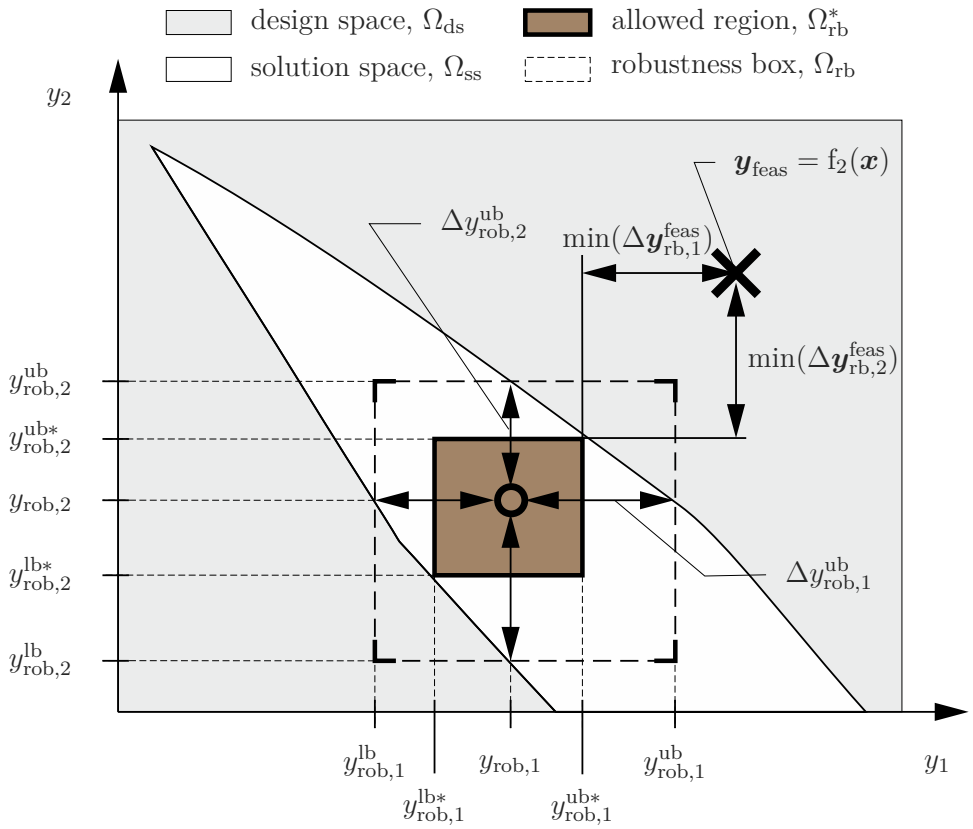


Figure 5.3: Robust Optimization for Target Cascading.

A two-dimensional example of a particular iteration step is shown in Figure 5.3. As f_{scale} is equal 0.5, the intervals of the robustness box $I_{rob,j} = [y^{lb}_{rob,j}, y^{ub}_{rob,j}]$ are scaled to the half of their width. In order to compute the shortest distance between the scaled solution box and the feasible design, the distance between all of the edges of the box and the feasible design must be calculated for a particular dimension first, as shown in Equation (5.8).

$$\Delta \boldsymbol{y}^{feas}_{rb,j} = \left[\frac{y_{feas,j} - y^{lb*}_{rob,j}}{y_{ds,j}}, \frac{y^{ub*}_{rob,j} - y_{feas,j}}{y_{ds,j}} \right] \tag{5.8}$$

Then, based on the classifier c_j, it is distinguished for each dimension whether the feasible design lies between two edges of the scaled robustness box such that the requirement is fulfilled, or not. Following Equation (5.8), the particular requirement is fulfilled, if both entities of $\Delta \boldsymbol{y}^{feas}_{rb,j}$ are negative.

In general, the goals of the robustness optimization within the target cascading procedure are:

1. identifying a robust design s.t. the scaled robustness box contains the feasible design identified in the previous iteration loop, $\boldsymbol{y}_{feas}(\boldsymbol{x}) \in \Omega^*_{rb}$,
2. maximizing the box size measure μ_{rb}.

Therefore, a penalty is only added for those dimensions in which the feasible design is located outside of the scaled robustness interval, which is distinguished by the classifier c_j

as shown in Equation (5.9).

$$c_j = \begin{cases} 1 & \text{if } \min\left(\Delta \boldsymbol{y}_{\mathrm{rb},j}^{\mathrm{feas}}\right) < 0 \\ 0 & \text{if } \min\left(\Delta \boldsymbol{y}_{\mathrm{rb},j}^{\mathrm{feas}}\right) > 0 \end{cases} \tag{5.9}$$

The penalty added in the performance function presented by Equation (5.12) depends on the distance between the scaled robustness box and the feasible design. Therefore, the Euclidean norm (also called 2-norm) is used, see Equation (5.10).

$$\left\| \min\left(\Delta \boldsymbol{y}_{\mathrm{rb},j}^{\mathrm{feas}}\right) c_j \right\| = \sqrt{\sum_{j=1}^{d} \left| \min\left(\Delta \boldsymbol{y}_{\mathrm{rb},j}^{\mathrm{feas}}\right) c_j \right|^2} \tag{5.10}$$

Due to the classifier c_j, only those dimensions where the feasible design is located outside of the scaled robustness interval are considered for the penalty. In order to identify the minimum distance between the j-th robustness interval and the feasible design, the minimum distance to both threshold values of the robustness interval is computed.

The general mathematical problem statement reads as follows: For a given design space $\Omega_{\mathrm{ds}}^{\mathrm{y}}$, a feasible design $\boldsymbol{y}_{\mathrm{feas}}$, a condensed performance function $\varphi_{\mathrm{rb}}(\Omega_{\mathrm{rb}})$ and a target region of the vehicle performance $\Omega_{\mathrm{sb}}^{\mathrm{z}}$ seek $\boldsymbol{y}_{\mathrm{rob}}$ such that

$$\begin{aligned} \text{if } \boldsymbol{y}_{\mathrm{feas}} \notin \Omega_{\mathrm{rb}}^{*} \text{ then} \quad & \|\overline{\Omega_{\mathrm{rb}}^{*} \boldsymbol{y}_{\mathrm{feas}}}\| \quad \rightarrow \min \\ \text{elseif } \boldsymbol{y}_{\mathrm{feas}} \in \Omega_{\mathrm{rb}}^{*} \text{ then} \quad & \mu_{\mathrm{rb}} \quad \rightarrow \max \\ \text{subject to} \quad & \boldsymbol{y}_{\mathrm{rob}} \in \Omega_{\mathrm{ss}}^{\mathrm{y}}. \end{aligned} \tag{5.11}$$

Therefore, it is necessary to ensure better ratings for the designs inside of the solution space compared to the ones outside of it in all cases – independent of whether the scaled robustness box contains the feasible design or not.

Resulting from the previous considerations, the exact problem statement for the robust optimization within the target cascading procedure reads:

$$\underset{\boldsymbol{y}_{\mathrm{rob}}}{\text{minimize}}\ \varphi_{\mathrm{rb}}(\Omega_{\mathrm{rb}})$$

$$\text{with} \quad \varphi_{\mathrm{rb}} = \begin{cases} -\ \mu_{\mathrm{rb}} + \left(\sqrt{\sum_{j=1}^{d} \left| \min\left(\Delta \boldsymbol{y}_{\mathrm{rb},j}^{\mathrm{feas}}\right) c_j \right|^2} - d \right) w_{\mathrm{pos}} & \begin{array}{l} \text{if } \max\left(\Delta \boldsymbol{y}_{\mathrm{rob}}^{\mathrm{lb}}\right) < 0 \\ \text{and } \min\left(\Delta \boldsymbol{y}_{\mathrm{rob}}^{\mathrm{ub}}\right) > 0 \end{array} \\ \sum_{j=1}^{n} \frac{\left|y_{\mathrm{rob},j}^{\mathrm{lb}}\right| + \left|y_{\mathrm{rob},j}^{\mathrm{ub}}\right|}{y_{\mathrm{ds},j}^{\mathrm{ub}} - y_{\mathrm{ds},j}^{\mathrm{lb}}} & \begin{array}{l} \text{if } \max\left(\Delta \boldsymbol{y}_{\mathrm{rob}}^{\mathrm{lb}}\right) \geq 0 \\ \text{or } \min\left(\Delta \boldsymbol{y}_{\mathrm{rob}}^{\mathrm{ub}}\right) \leq 0 \end{array} \end{cases}$$

$$\text{and} \quad \mu_{\mathrm{rb}} = \prod_{j=1}^{d} \frac{\Delta y_{\mathrm{rob},j}^{\mathrm{lb}}\, \Delta y_{\mathrm{rob},j}^{\mathrm{ub}}}{y_{\mathrm{ds},j}^{\mathrm{ub}} - y_{\mathrm{ds},j}^{\mathrm{lb}}} w_j. \tag{5.12}$$

Depending on whether the design to be evaluated $\boldsymbol{y}_{\mathrm{rob}}$ is included in the solution space or not, the *condensed vehicle performance* φ_{rb} is computed differently.

If the solution space does not contain the design, the performance is measured by the sum of distances to the solution space in each dimension normalized to the design space, see the *lower case* of Equation 5.12. Therefore, the performance of a design outside of the solution space is always positive.

If the solution space contains the design, the vehicle performance depends on two different parts, see the *upper case* of Equation 5.12. The first part of the expression consists of the negative *box size measure* μ_{rb} which evaluates the size of the robustness box, as introduced in Subsection 2.3.3. Therefore, a larger robustness box results in better performance. The second part of the expression evaluates the distance between the scaled robustness box and the feasible design. Hence, a penalty is added if the scaled robustness box does not contain the feasible design. As described by Equation 5.10, only those dimensions with a scaled robustness interval that does not contain the related *subsystem performance* $y_{\text{feas},i}(\boldsymbol{x})$ of the feasible design are considered for the penalty. To clearly distinguish between designs included in the solution space and those which are not, the condensed performance of designs included in the solution space need to be negative.

How to ensure better ratings for all possible designs within the solution space compared to the ones outside of it. In order to avoid a positive value of the condensed performance function in the upper case of Equation 5.12, the number of dimensions is subtracted as an offset value. Since the Euclidean norm of the number of dimensions is always smaller compared to the sum of them, the condensed performance is always negative if the design is included in the solution space. In practice, it is not the case that the distance between the robustness interval and the feasible design gets as big as the associated design interval simultaneously for all dimensions. Therefore, the design would need to be located in the corner of the design space and the robustness box would need to be infinitely small, while the feasible design is located in the opposite corner of the design space. Usually, this is not the case as long as the design space is selected properly.

The importance of the feasibility condition is adjusted by the *weighting factor* w_{pos}. Usually, the importance of feasibility is adjusted much higher than the importance of the box size measure. Therefore, it is ensured that if the scaled robustness box does not include the feasible design, its boundaries are as close as possible to the feasible design. Through the aid of this clear prioritization, convergence of the entire method is ensured. Nevertheless, the scaling factor provides the possibility to adjust the importance whenever feasibility is not that important. The scaling factor f_{scale}, which scales the robustness box Ω_{rb} to Ω_{rb}^{*}, defines the allowed distance between feasible design and robustness box. Therefore, it determines how hard the feasibility criterion may be reached.

5.2.2 Extension of the Solution Box Optimization for Target Cascading.

For the automatic target cascading algorithm, the solution box optimization needs to be extended in order to take the feasibility restrictions on the detail level into account.

The overall goals of the solution box optimization within the target cascading procedure are:

1. identifying a solution box which contains the feasible design of the previous iteration loop $\boldsymbol{y}_{\text{feas}}(\boldsymbol{x}) \in \Omega^{\text{y}}_{\text{sb}}$,
2. maximizing the box size measure μ_{sb}.

In addition, the solution box contains good designs only with the exception of a percentage of allowed bad designs $B_{\%,\text{crit}}$.

In general, the mathematical problem statement reads as follows: for a given design space $\Omega^{\text{y}}_{\text{ds}}$, a feasible design $\boldsymbol{y}_{\text{feas}}$, a condensed performance function $\varphi_{\text{sb}}(\Omega_{\text{sb}})$ and a target region of the vehicle performance $\Omega^{\text{z}}_{\text{sb}}$, seek $\Omega^{\text{y}}_{\text{sb}}$ such that

$$\begin{aligned} \text{if } \boldsymbol{y}_{\text{feas}} \notin \Omega^{\text{y}}_{\text{sb}} \text{ then} \quad & \|\overline{\Omega^{\text{y}}_{\text{sb}} \boldsymbol{y}_{\text{feas}}}\| \quad \rightarrow \min \\ \text{elseif } \boldsymbol{y}_{\text{feas}} \in \Omega^{\text{y}}_{\text{sb}} \text{ then} \quad & \mu_{\text{sb}}(\Omega^{\text{y}}_{\text{sb}}) \quad \rightarrow \max \\ \text{subject to} \quad & B_{\%}(\Omega^{\text{y}}_{\text{sb}}, \Omega^{\text{y}}_{\text{ss}}) \leq B_{\%,\text{crit}}. \end{aligned} \tag{5.13}$$

Depending on both cases, it is not necessary to separate the expansion strategy of the solution box. Starting from the robust design, which was identified in stage 1 of the automatic algorithm, the solution box grows in each direction at the beginning of the optimization, see Figure 5.2. Nevertheless, the solution box receives a greater reward for growing in the direction of the feasible design. Based on a specified weighting between the size of the solution box and its feasibility, a trade-off is defined between both.

In order to consider the feasibility of the solution box, an additional term is added to the performance function, which was introduced in Subsection 2.3.3. Therefore, the *condensed vehicle performance* φ_{sb} of a particular solution box depends on the *box size measure of the solution box* μ_{sb}, the *feasibility measure of the solution box* φ_{feas} as well as its percentage of bad designs $B_{\%}$.

The exact mathematical problem statement for the solution box optimization within the target cascading procedure reads:

$$\begin{aligned} &\underset{\mathbf{y}^{\text{lb}},\mathbf{y}^{\text{ub}}}{\text{minimize}} \quad \varphi_{\text{sb}}(\Omega_{\text{sb}}) \\ &\text{with} \qquad \varphi_{\text{sb}} = \begin{cases} - \quad \mu_{\text{sb}} - \varphi_{\text{feas}}\, w & \text{if } B_{\%} < B_{\%,\text{crit}} \\ - \quad \mu_{\text{sb}} - \varphi_{\text{feas}}\, w + p & \text{if } B_{\%} \geq B_{\%,\text{crit}} \end{cases} \\ &\text{and} \qquad p = (\mu_{\text{sb}} + \varphi_{\text{feas}}\, w)\ (B_{\%} - B_{\%,\text{crit}}) \\ &\text{subject to} \quad \mathbf{y}^{\text{lb}} \in \boldsymbol{I}^{\text{lb}}_{\text{ds}},\ \mathbf{y}^{\text{ub}} \in \boldsymbol{I}^{\text{ub}}_{\text{ds}} \end{aligned} \tag{5.14}$$

Therefore, by adjusting the design variables $\mathbf{y}^{\text{lb}}$ and $\mathbf{y}^{\text{ub}}$ the vehicle performance is minimized as a function of the solution box. The design variables are the permissible intervals on the y-level that may be shifted between the robust design of stage 1 and the associated design space boundaries, such that $\mathbf{y}^{\text{lb}} \in \boldsymbol{I}^{\text{lb}}_{\text{ds}}$ and $\mathbf{y}^{\text{ub}} \in \boldsymbol{I}^{\text{ub}}_{\text{ds}}$. The vehicle performance depends on the size measure of the solution box as well as its feasibility. A trade-off between both is adjusted by the weighting factor w. If the part of Equation

5.14 that evaluates the feasibility is significantly stronger than the box size measure, the solution box optimization will preferably shift its boundaries towards the feasible design. If the solution box exceeds the allowed percentage of bad designs $B_{\%} \geq B_{\%,\mathrm{crit}}$ a positive *penalty* p is added to the performance function. Since the penalty $p(\mu_{\mathrm{sb}}, \varphi_{\mathrm{sb}}, B_{\%})$ is also a function of the box size measure and the feasibility, the solution box is prevented from growing too far out of the solution space. Thus, a solution box with a better rating regarding its size and feasibility automatically obtains an increased penalty while growing too far out of the solution space.

The feasibility measure φ_{feas} depends on the distance between the solution box and the feasible design. It reaches its absolute maximum as soon as the feasible design, which was identified in stage 3 of the previous iteration loop, is part of the solution box. From then on, the reward for feasibility is capped and the solution box grows in each direction depending on the box size measure, $\mu_{\mathrm{sb}} \rightarrow \max$. For good optimization results, it is recommended to select the importance of the feasibility term significantly higher than the size measure μ_{sb}. By adding the box size measure to the feasibility term, the condensed performance of the vehicle/solution box φ_{sb} is computed.

The feasibility of a particular solution box is evaluated by

$$\varphi_{\mathrm{feas}} = \frac{\prod_{j=1}^{t} \varphi_{\mathrm{feas},j}}{t}, \tag{5.15}$$

where $\varphi_{\mathrm{feas},j}$ evaluates the feasibility of a particular permissible interval, with $j = 1,2,\ldots,t$. Since not all permissible intervals on the y-level are subject to feasibility constraints, the number of dimensions t considered for the feasibility evaluation is usually smaller than the total number of dimensions d. For each design variable that is considered as an objective criterion in the feasibility optimization on the detail level, the performance reward is computed according to Equation (5.16).

$$\varphi_{\mathrm{feas},j} = \begin{cases} \min\left(y_j^{\mathrm{ub}} - y_{\mathrm{rob},j}, \Delta y_{\mathrm{rob},j}^{\mathrm{feas}}\right) & \text{if } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} > 0 \\ \min\left(y_{\mathrm{rob},j} - y_j^{\mathrm{lb}}, \left|\Delta y_{\mathrm{rob},j}^{\mathrm{feas}}\right|\right) & \text{if } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} < 0 \end{cases} \tag{5.16}$$

An example of an intermediate iteration loop is shown in Figure 5.4 as an illustration of the solution box evaluation. Therefore, the previously computed robust design and feasible design are depicted as well. The distance between feasible and robust design in a particular dimension is computed by $\Delta y_{\mathrm{rob},j}^{\mathrm{feas}} = y_{\mathrm{feas},j} - y_{\mathrm{rob},j}$. Both designs result from the previous design stages – stage 1 of the current and stage 3 of the previous iteration loop. Hence, the distance between both does not change in stage 2. Due to the minimum function in Equation (5.16), the feasibility reward for each permissible interval is capped if feasibility is established. A particular permissible interval of the solution box is called feasible if it includes the closest feasible design of stage 3, as described at the beginning of Section 5.2. In order to evaluate whether a permissible interval is feasible or not, it is assessed if the distance between the feasible design and the robust design is greater than the distance between the boundaries of the permissible interval and the robust design. Therefore, it must be distinguished whether the particular property of the feasible design

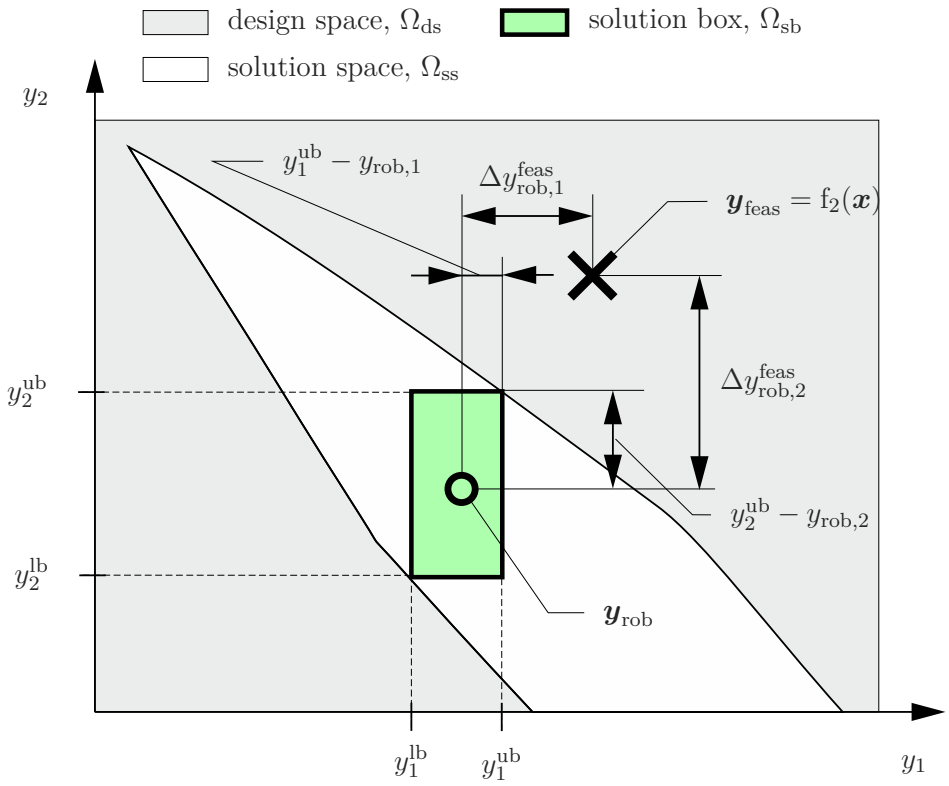


Figure 5.4: Solution Box Optimization for Target Cascading.

is higher ($\Delta y_{\mathrm{rob},j}^{\mathrm{feas}} > 0$) or lower ($\Delta y_{\mathrm{rob},j}^{\mathrm{feas}} < 0$) than the associated one of the robust design. To compute the distance between a permissible interval and the robust design, the critical bound needs to be identified. By definition, the critical bound is the bound in the direction in which the robust design needs to extend in order to ensure feasibility. Therefore, the upper bound is used if the distance between feasible and robust design is greater than zero, otherwise the lower bound is used. For the complete case distinction, see Equation 5.17.

$$\text{permissible interval is feasible} = \begin{cases} 0 & \text{if } y_j^{\mathrm{ub}} - y_{\mathrm{rob},j} < \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} \quad \text{and } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} > 0 \\ 1 & \text{if } y_j^{\mathrm{ub}} - y_{\mathrm{rob},j} > \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} \quad \text{and } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} > 0 \\ 0 & \text{if } y_{\mathrm{rob},j} - y_j^{\mathrm{lb}} < \left|\Delta y_{\mathrm{rob},j}^{\mathrm{feas}}\right| \quad \text{and } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} < 0 \\ 1 & \text{if } y_{\mathrm{rob},j} - y_j^{\mathrm{lb}} > \left|\Delta y_{\mathrm{rob},j}^{\mathrm{feas}}\right| \quad \text{and } \Delta y_{\mathrm{rob},j}^{\mathrm{feas}} < 0 \end{cases} \tag{5.17}$$

For the j-th design variable, the permissible interval is feasible if the distance between the j-th critical bound and the robust design $y_{\mathrm{rob},j}$ exceeds the *distance between feasible and robust design* $\Delta y_{\mathrm{rob},j}^{\mathrm{feas}}$.

5.2.3 A Simple Two Dimensional Example

The automatic algorithm described in the previous subsection is applied to a simple design problem in the field of vehicle dynamics first. Therefore, three objective criteria are considered on the z-level in order to ensure customer satisfaction, see Figure 5.5. Based

on the overall design objectives, permissible intervals on two functional subsystem properties of the subsystem level (y-level) are derived. Designs within the permissible intervals must satisfy all overall design objectives, but also must be feasible by adapting the design variables on the detail level (x-level). For the evaluation of particular designs, ANNs are used based on the vehicle model described in Subsection 2.2.2 and a kinematic suspension model further described in Chapter 4.

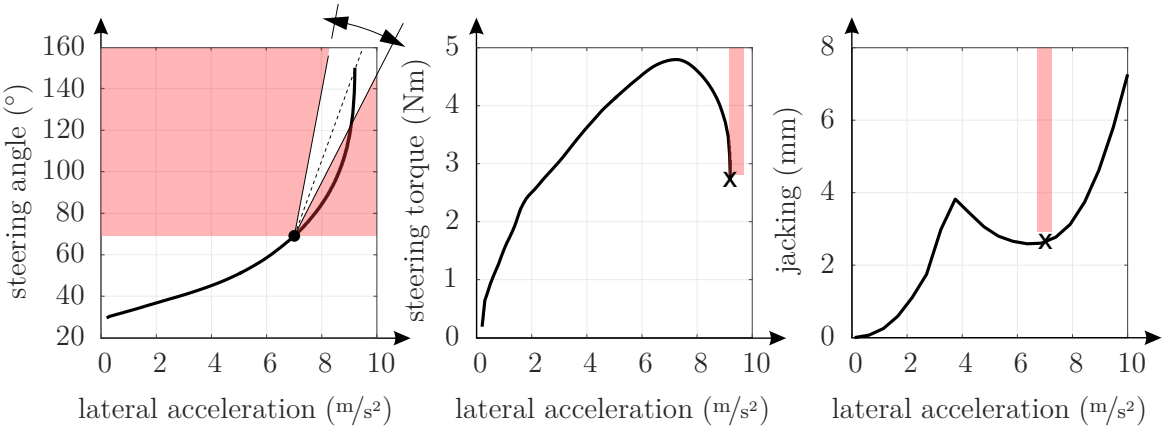


Figure 5.5: Vehicle dynamics example problem considering three targets on the overall system.

Objective and Design Variables

Parameters on the z-level. In order to ensure customer satisfaction, three objective criteria are applied to the overall vehicle behavior. The overall objective criteria on the vehicle are: the jacking of the vehicle body Δz_{CoG} at 7 $\mathrm{m/s^2}$ lateral acceleration, the understeer behavior in the area of lateral limit behavior $\Delta\delta_{\mathrm{h},95\%}$ and the manual torque of the driver during maximum lateral acceleration $M^{\mathrm{ub}}_{\mathrm{h,ay-max}}$. Jacking of the vehicle body is caused by nonlinearities in the characteristic curve of vertical wheel force over wheel travel, the suspension geometry (e.g. RC_{h}, gRC_{h}) or both. Since jacking feels inconvenient to the driver, its value should not exceed a critical upper threshold value $\Delta z^{\mathrm{ub}}_{\mathrm{CoG}}$. In order to enable driving pleasure in the area of maximum lateral acceleration without neglecting driving safety, $\Delta\delta_{\mathrm{h},95\%}$ needs to lie within a permissible interval as well. If its value exceeds the upper threshold value $\Delta\delta^{\mathrm{ub}}_{\mathrm{h},95\%}$, the car would be too indirect and hence cannot use its full potential in lateral dynamics. Nevertheless, if the lower threshold value $\Delta\delta^{\mathrm{lb}}_{\mathrm{h},95\%}$ is not reached, the stability of the vehicle is not sufficient. The manual torque applied by the driver during maximum lateral acceleration is not permitted to exceed the threshold value $M^{\mathrm{ub}}_{\mathrm{h,ay-max}}$, otherwise it is difficult to recognize the limit behavior.

Parameters on the y-level. On the y-level, the gradient of the roll center height variation over wheel travel $gRC_{\mathrm{h},FA}$ and the roll center height at construction level $RC_{\mathrm{h},FA}$ are allowed to be adapted in order to fulfill the requirements on the z-level. The roll center is the axis the vehicle body is rolling around during cornering. The higher the roll center the less roll angle occurs during cornering due to the decreasing lever-arm between roll center and CoG (as long as $RC_{\mathrm{h}} < h_{\mathrm{CoG}}$).

Conflicting Goals. The slightest jacking is achieved by putting the RC_{h} as low as possible, therefore jacking caused by the suspension geometry is reduced. Nevertheless, the $RC_{\mathrm{h},FA}$ should not be too low, as this results in an unacceptably high $M^{\mathrm{ub}}_{\mathrm{h,ay-max}}$ (without consideration of a steering torque controller). Since the understeer behavior is also influenced by the roll center height and their variation, selecting the right trade-off is even more complicated.

Parameters on the x-level. On the detail level (x-level), the suspension geometry needs to be adapted in order to fulfill the requirements on the subsystem performance (y-level). Since a modular construction kit should be used, only the z-positions of the outer kinematic joints of the upper A-arm f_{z} and lower control link h_{z} are variable.

An overview of the considered parameters is shown in Table 5.1.

Table 5.1: Overview of parameters on different abstraction levels.

Abstraction Level	**Description**	**Symbol**
z-level	jacking of the body on the front axle	Δz_{CoG}
	understeer behavior at 95 % of maximum lateral acceleration	$\Delta\delta_{\mathrm{h,95\%}}$
	steering wheel torque perceived by the driver at maximum lateral acceleration	$M_{\mathrm{h,ay-max}}$
y-level	roll center height at construction level at the front axle	$RC_{\mathrm{h},FA}$
	gradient of the roll center height variation over wheel travel at the front axle	$gRC_{\mathrm{h},FA}$
x-level	z-position of upper control link outer	f_{z}
	z-position of lower control link outer	h_{z}

Results and Evaluation of the Algorithm

Regardless whether the initial design is good or bad, the algorithm starts with a robust optimization in stage 1. The first robust design $\boldsymbol{y}_{\mathrm{rob}}$ is found by maximizing the distance to the closest bad designs, as shown in Iter0 of Figure 5.6 ($\mu(\Omega_{\mathrm{rb}}) \rightarrow \max$). Afterward, the solution box optimization (stage 2) finds the solution box with a maximum box size measure including the robust design followed by stage 1 ($\mu(\Omega_{\mathrm{sb}}) \rightarrow \max$ and $\boldsymbol{y}_{\mathrm{rob}} \in \Omega_{\mathrm{sb}}$).

Subsequently, according to Section 4.1, the feasibility optimization (stage 3) finds the feasible design $\boldsymbol{y}^{\mathrm{opt}}_{iter}$ with the shortest distance to the solution box followed by stage 2. In the following iterations (iter1 ... iter6), the solution box and the closest feasible design converge rapidly to each other until a feasible solution box is found.

Figure 5.7 summarizes the iterations of the algorithm. In (a) a glance on the y-level shows that the largest solution box is not overlapping the feasible space. Although the z-

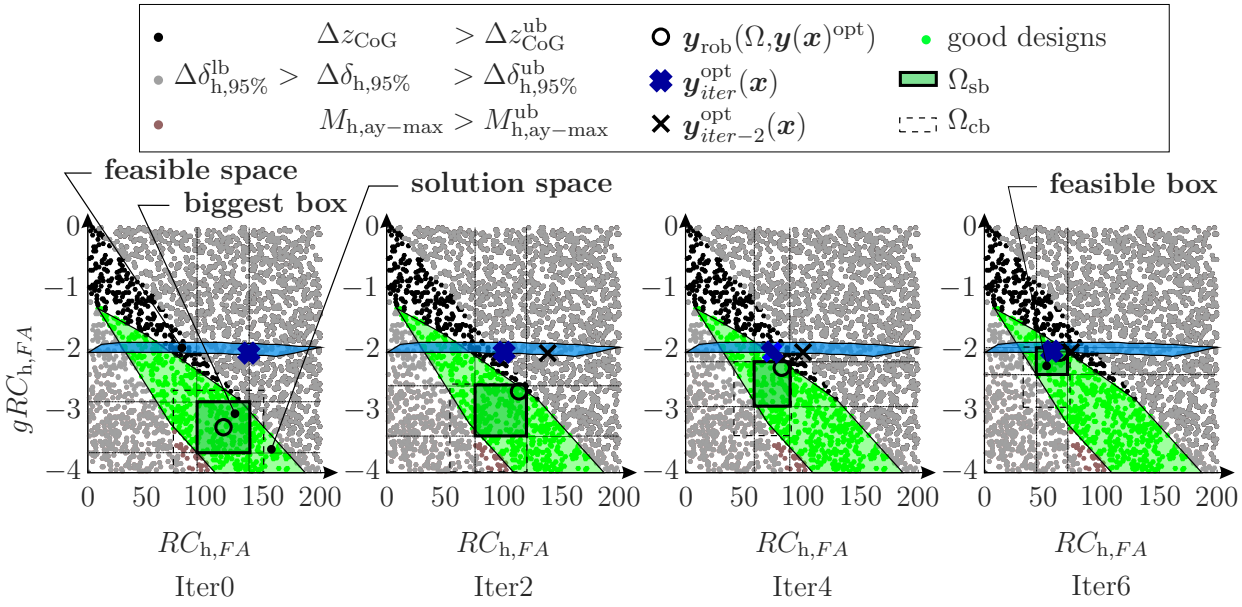
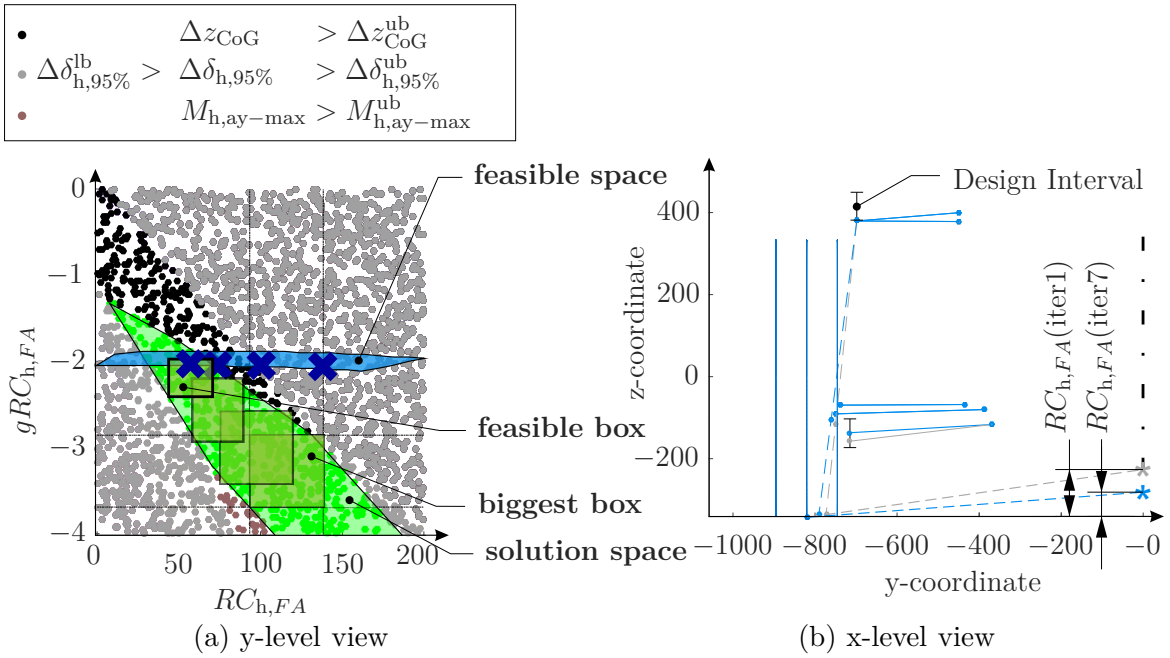


Figure 5.6: Evolution of solution box and feasible design on the functional subsystem level (y-level).

position of the two kinematic joints may be adapted in order to satisfy the requirement on $RC_{\mathrm{h},FA}$, the requirement on $gRC_{\mathrm{h},FA}$ is failed. The reason for that lies in the low sensitivity of the z-positions of the adaptable ball joints on the z-variation of the roll center during wheel travel. Since the z-variation mainly depends on the variation of the instantaneous center of the wheel carrier, the length of the control links would need to be changed in order to satisfy the requirement.

In this case, the distance in y-direction between the inner and outer joints is not allowed to be changed due to packaging requirements. Therefore, the solution box must be shifted in order to ensure feasibility. To do so, the solution box shrinks from its initial iteration (iter0) to a box that is overlapping the feasible space (iter6). In (b) the geometrical design of the suspension at iter0 (gray) and iter7 (blue) is shown.

Figure 5.8 shows five *key performance indicators* (KPIs) for evaluating the performance of the algorithm (φ_{feas}, φ_{sb}, φ_{rb}, μ_{sb}, $B_{\%}$). The set of design parameters for each of the three stages is evaluated by a separate performance criterion φ. The performance of the Robust Optimization (Subsection 2.3.3) is evaluated by φ_{rb}, which reaches its performance peak in iteration step 0, since no feasible design is known, and therefore the largest candidate box is found. The following iteration steps show worse performance due to the large distance to the closest feasible design. Therefore, the robust optimization reduces this distance first, and then maximizes the box size measure of the robustness box. The performance of the solution box optimization (Subsection 2.3.4) is evaluated by φ_{sb}. As expected, its value is positively correlated with the performance value of the robust optimization. On one hand, the most robust design offers the greatest potential for a solution box with maximum volume μ_{sb}, and on the other hand, if the preceding robust design is far away from the feasible space, the best solution box within the current iteration step is far away as well.



(a) y-level view (b) x-level view

Figure 5.7: Two dimensional example on different abstraction levels.

The feasibility of the requirements is evaluated based on the feasibility optimization by the performance criterion φ_{feas} that was presented in Section 4.1.

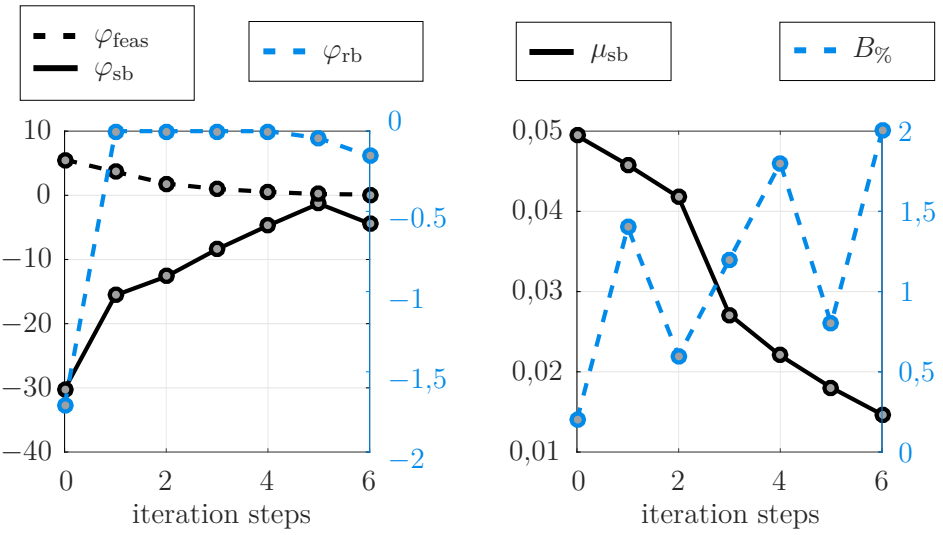


Figure 5.8: KPIs of the automatic target cascading algorithm using solution spaces.

As the requirements become increasingly feasible, φ_{feas} gets smaller after each iteration. Two additional KPIs μ_{sb} and $B_{\%}$ may be used to monitor the success of the overall algorithm. The steady downward trend of the box size measure μ_{sb} shows that the largest solution box was found under the given constraints within each iteration, while the constraints were tightened after each feasibility optimization. The percentage of failed designs $B_{\%}$ within the best solution box in each iteration varies between zero and two percent,

which is acceptable when taking the selected critical value of one percent into account.

5.3 Semi-Automatic Algorithm

Comparison between automatic and semi-automatic algorithm.

Compared to the automatic algorithm within the semi-automatic algorithm, the engineer is an integral part of the overall optimization procedure.

Here the engineer is responsible for:

1. Shifting the solution box towards the feasible design.

2. Ensuring that the fraction of good designs within the solution box is sufficient.

5.3.1 Methodology

For identifying a feasible solution box, the *semi-automatic algorithm* is divided into two phases. Phase 1, identifies the solution box Ω_{sb} with the maximum box size measure $\mu(\Omega_{\text{sb}})$. Phase 2 determines a feasible solution box, see Figure 5.9. Both phases generally consist of two steps. Phase 1 applies the same tools that were described in Section 2.3 for the vehicle architecture design. A novel iterative search algorithm based on numerical optimization, knowledge management and the decision making engineer/user is applied in phase 2. Therefore, in each iteration, a feasible and robust design is sought according to Section 4.1 (step 1). In the second step, the current solution box is adapted with respect to feasibility and top-level target fulfillment based on prepared information of the previous steps in an iterative manner. The iterative search algorithm is stopped manually by the user if the current solution box satisfies his requirements on the size measure as well as the requirements on robustness and feasibility.

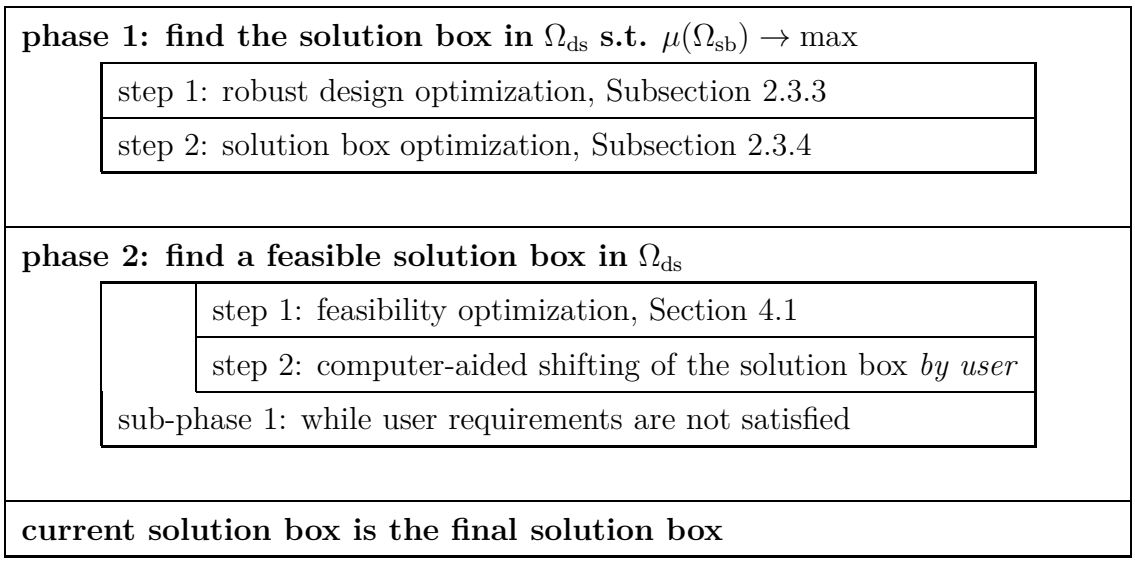


Figure 5.9: Phases of the semi-automatic algorithm for identifying a feasible solution box.

Adapting the position and size of the biggest solution box (step 2 of phase 2) is necessary as long as not all permissible intervals are satisfied by the performance of the subsystem. Therefore, in general, the solution box is shifted towards the direction of the worst feasible permissible interval and then adapted manually by the user. The entire procedure of the shifting algorithm consists of three nested loops and is presented in Figure 5.10.

Input data for the computer-aided shifting of the solution box is the solution box Ω_{sb} from the previous iteration as well as the performance of the optimized subsystem y_{feas}. The outer loop is applied to all permissible intervals that are not feasible by the optimized subsystem performance starting with the least feasible one. The least feasible permissible interval is defined as the one with the largest normalized[7] distance to the performance of the subsystem. At the beginning of the outer loop, the qualifier $feasibility\ optimization$, which decides whether a feasibility optimization should be performed, is set to zero. Thereafter, the **if** statement in Figure 5.10 checks whether the j-th not feasible permissible interval is defined as a flexible interval or not. The user has the possibility to predefine flexible intervals in order to exclude non-flexible intervals from the solution search, which makes sense if an existing modular system should be used, for example.

If the failed interval is defined as a flexible interval, it is shifted automatically by the algorithm such that its closest boundary overlaps the performance of the subsystem. As the biggest solution box is restricted by bad designs, a displacement results in an increased fraction of bad designs. Therefore, the fraction of bad designs is computed for each objective criterion and comprised in the vector $\boldsymbol{z}_{\%}^{\text{bad}}$. The **while** statement is applied as long as the fraction of bad designs caused by a particular objective criterion remains above a user-defined threshold value $z_{\%,\text{crit}}^{\text{bad}}$. The inner loop is applied to the most sensitive design variables. Its goal is to identify the design variables that are suited to reduce the fraction of bad designs by shifting the boundaries of their associated permissible intervals. Therefore, the correlation coefficients R_j and R_k may be used to distinguish between direct proportional and inverse proportional design variables. While R_j estimates the causal relations between the j-th not feasible permissible interval and the objective criterion $\max(\boldsymbol{z}_{\%}^{\text{bad}})$, R_k estimates the causal relations between the k-th most sensitive design variable and $\max(\boldsymbol{z}_{\%}^{\text{bad}})$. Based on the conditions stated in Table 5.2, for proportional relations between R_j and R_k, one out of four categories is assigned to each design variable.

Table 5.2: Categories for proportional compensation variables.

proportional	$y_{\text{norm},k} < -0.5$	$-0.5 < y_{\text{norm},k} < 0$	$0 < y_{\text{norm},k} < 0.5$	$0.5 < y_{\text{norm},k}$
$y_{\text{feas},j} < y_j^{\text{lb}}$	unsuitable	3rd order	2nd order	1st order
$y_{\text{feas},j} > y_j^{\text{ub}}$	1st order	2nd order	3rd order	unsuitable

[7]The distance is normalized with respect to the size of the associated permissible interval.

data: solution box Ω_{sb}, subsystem performance after feasibility optimization $\boldsymbol{y}_{\mathrm{feas}}$
for j = 1,2,..., number of permissible intervals that are not feasible
  set qualifier for $feasibility\ optimization = 0$
  if $I_j^{\mathrm{y}} \rightarrow$ flexible interval
    shift I_j^{y} such that $\min(|y_j^{\mathrm{ub}} - y_j|, |y_j^{\mathrm{lb}} - y_j|) = 0$
    compute the fraction of bad designs that violate overall design targets on z-level within the new solution box Ω_{sb}
    while $\max(\boldsymbol{z}_{\%}^{\mathrm{bad}}) > z_{\%,\mathrm{crit}}^{\mathrm{bad}}$ **then**
      sort all design variables on their sobol indices in descending order
      for k = 1,2,..., number design variables that should be considered
        if correlation coefficients R_j **and** R_k have the same sign **then**
          y_j and y_k are direct proportional
          assign compensation category to the design variable
        else
          y_j and y_k are inverse proportional
          assign compensation category to the design variable
        end
      end
      $\Rightarrow$ *user*: adapt compensation parameters in order to satisfy $z_{\%,\mathrm{crit}}^{\mathrm{bad}}$
      reevaluate the solution box Ω_{sb}
    end
    set qualifier for $feasibility\ optimization = 1$
    break
  elseif $I_j \nrightarrow$ flexible interval **and** j = number of failed permissible intervals
    adapt the permissible intervals that are hardly feasible in order to relax feasibility and satisfy overall vehicle requirements
    set qualifier for $feasibility\ optimization = 1$
    break
  end
end
result: new candidate solution box Ω_{sb}

Figure 5.10: Algorithm to balance requirements.

Therefore, the user is supported with the following information for manual shifting of the solution box:

- information about the position of the current solution box and closest feasible design visualized by projections, MPPs and areas for characteristic curves, see Subsection 2.3.5,
- information about the sensitivity of preselected design variables, and,
- information about the compensation category of the design variables.

After all necessary adaptations are made, the new solution box is re-evaluated. If the fraction of bad designs no longer exceeds the critical threshold value, the qualifier for the feasibility optimization is set to one.

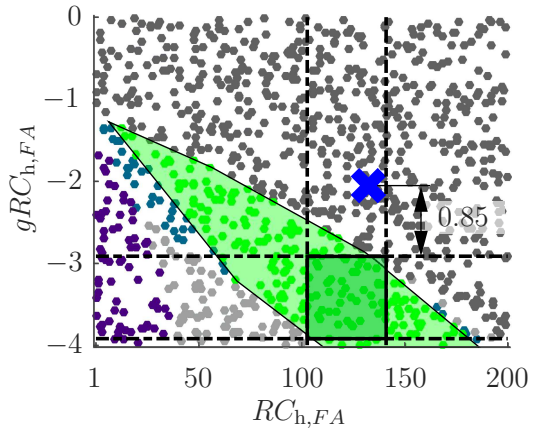
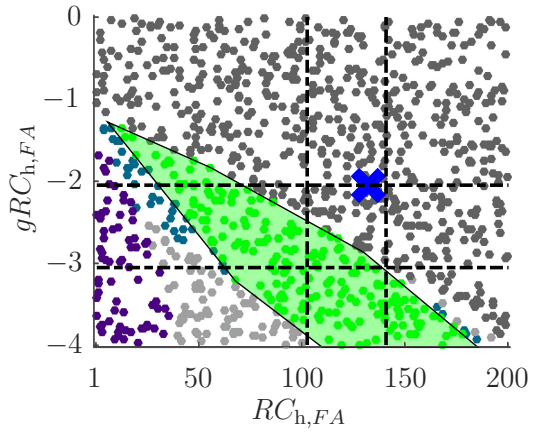
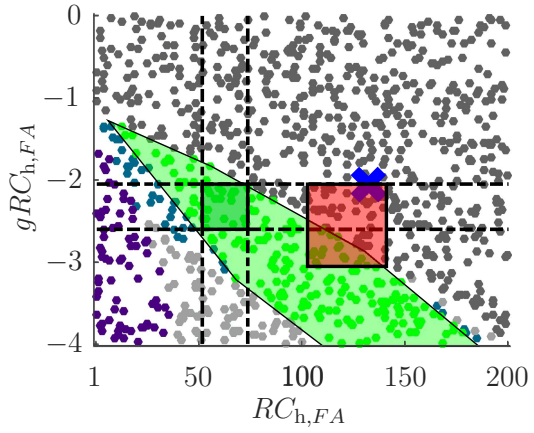
In some cases, it may happen that only one permissible interval is not feasible while it is also defined as not flexible. In this case, the user needs to identify the flexible and feasible variables that are hardly feasible in order to shift their permissible intervals and relax feasibility. Therefore, the normalized MPP is suited for identifying design variables with critical feasibility as the performance of the optimized subsystem is presented in relation to the permissible intervals. The difficultly of the realization of a particular permissible interval is measured by the minimum absolute distance between the bounds of the permissible interval and the subsystem performance. When the user has made his adoptions, the qualifier for the feasibility optimization is set to one.

The result of step 2 of phase 2 is a new candidate solution box whose feasibility is reevaluated in step 1 of phase 2 of the following iteration, see Figure 5.9. Phase 2 repeats until either the user is satisfied with the size of the solution box and its feasibility or the optimization is stopped manually.

5.3.2 A Simple Two Dimensional Example

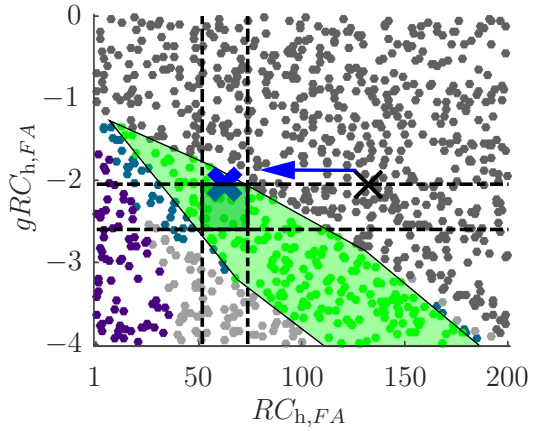
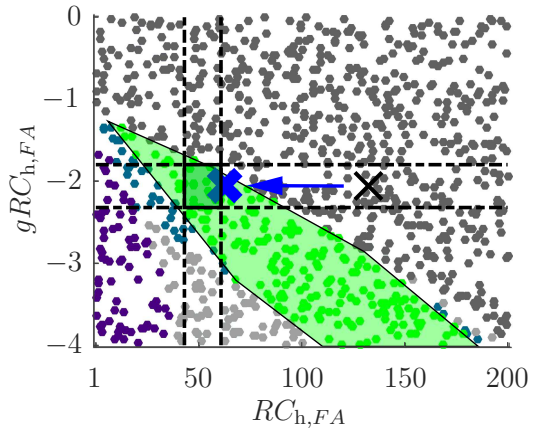
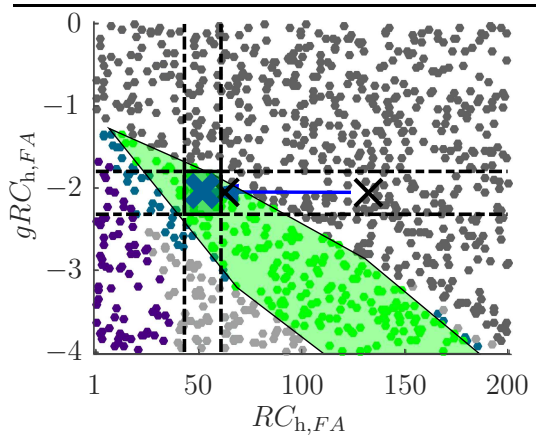
For comparing the introduced algorithms, the semi-automatic algorithm is applied to the same example that was described in Subsection 5.2.3. Therefore, $\boldsymbol{y} = [RC_{\mathrm{h},FA}, gRC_{\mathrm{h},FA}]$ describes the two design variables on the y-level. Although the semi-automatic algorithm is only applied to a two-dimensional example, the same procedure is also applicable to design problems with more design variables.

Table 5.3: A two dimensional example of the Semi-Automatic Algorithm.

Parameter Space	Description
	**Iteration 0** **0.1 solution box optimization:** Initially the solution box with the maximum size measure is identified according to Section 2.3. **0.2 feasibility optimization:** In the following, a feasibility optimization leads to the closest feasible design on the y-level, as described in Section 4.1.
	**Iteration 1** **1.1 automated shifting of the solution box:** As the requirements of $RC_{\mathrm{h},FA}$ and $gRC_{\mathrm{h},FA}$ imposed by the biggest solution box are not feasible by adapting the suspension design, the solution box needs to be shifted to establish feasibility. Shifting the solution box considering feasibility may be automated. Since the greatest normalized distance between Ω_{sb} and $\boldsymbol{y}_{\mathrm{feas}}$ is in the direction of $gRC_{\mathrm{h},FA}$ with a value of *0.85*, the solution box is shifted towards $\boldsymbol{y}_{\mathrm{feas}}$. As a result, the ratio between bad and good designs increases.
	**Iteration 1** **1.2 manual shifting of the solution box:** However, automatic shifting of the solution box for establishing feasibility does not take requirements satisfaction into account. Therefore, the fraction of bad designs regarding the threshold value $\Delta z_{\mathrm{CoG}}^{\mathrm{ub}}$ is increased by 66%. The user now manually needs to adapt the permissible intervals of the solution box. Therefore, the semi-automatic algorithm proposes design variables which are suitable for reducing the fraction of bad designs. By combining the information about sensitive design variables with the information about regions in the solution space that are not feasible, the user is guided to a feasible solution box.

Continued on next page

Table 5.3 – continued from previous page.

Parameter Space	Description
	**Iteration 1** **1.3 feasibility optimization:** Another feasibility optimization evaluates whether the requirements resulting on the suspension system from the adapted solution box are feasible or not. In the presented case, the resulting feasible design is slightly outside of the solution box.
	**Iteration 2** **2.1 & 2.2 shifting of the solution box:** As the feasible design is already very close to the solution box, the adoption of the permissible intervals through automatic shifting may be neglected. Since the user usually is not satisfied with a solution box where the most robust and feasible design is on the borderline of the solution box, it is adapted again in order to shift $y_{\mathrm{feas}},2$ closer to the center of the solution box.
	**Iteration 2** **2.3 feasibility optimization:** The following feasibility optimization identifies a robust design considering the updated permissible intervals. If the user decides that the feasible design is close enough to the center of the solution box and the solution box also is big enough, he stops the optimization.
legend • $\Delta z_{\mathrm{CoG}} > \Delta z^{\mathrm{ub}}_{\mathrm{CoG}}$ • $\Delta\delta^{\mathrm{lb}}_{\mathrm{h},95\%} > \Delta\delta_{\mathrm{h},95\%} > \Delta\delta^{\mathrm{ub}}_{\mathrm{h},95\%}$ • $M_{\mathrm{h,ay-max}} > M^{\mathrm{ub}}_{\mathrm{h,ay-max}}$ ✖ $\boldsymbol{y}^{\mathrm{opt}}_{iter}(\boldsymbol{x})$ ▭ Ω_{sb}	

CHAPTER 6

APPLICATION

In this chapter, the methods and tools described in the previous chapters are applied to practical design problems of vehicle architectures. Therefore, a new vehicle architecture should be designed for the next generation of luxury cars in order to meet the requirements of the market on, e.g., vehicle dynamics, electrification and fuel consumption. In the following examples, a wide range of requirements on vehicle dynamics as well as other conflicting requirements is considered. The objectified overall targets on the vehicle are derived based on:

- *Subjective evaluations of reference vehicles* from which the objectified overall vehicle performance is known.
- *Preferred subsystem configurations* of the manufacturer, such as a particular type of steering system or a set of tires. The preference may be caused by part costs, batch size and commonality
- *Expert knowledge* is used to consider requirements on the product that are not objectified or where the relations between objectives and design variables are not quantified.

Therefore, the tires, suspension and steering system may need to be adapted in order to satisfy the overall vehicle targets. Since the design variables of the subsystems and the objective criteria of the vehicle are highly interlinked with each other, considering all of them during the design process is challenging. The first example will show how to cope with this challenge using the example of suspension design.

Design Scenario 1. The successor of an existing vehicle should be developed. The main objective is to reduce the maximum force occurring on the steering rack in order to use a convenient EPS. Nevertheless, all other customer relevant properties should be at least as good as those of the predecessor. Due to changes to the drivetrain topology, a new

suspension system needs to be designed. However, the tires are considered to be the same, since no great improvements that result in lower steering rack forces are expected. In the lower right-hand corner of Figure 6.1, it can be seen that good designs exist only if the EPS provides a maximum steering rack force around 15 000 N. In the upper center of Figure 6.1 another possibility to satisfy vehicle requirements by reducing the variation of the kinematic trail until the maximum steering wheel angle gets clear is depicted. However, the design process will show that adapting $\Delta b_{\max}$ only is not feasible by optimizing the suspension system and how to handle this issue.

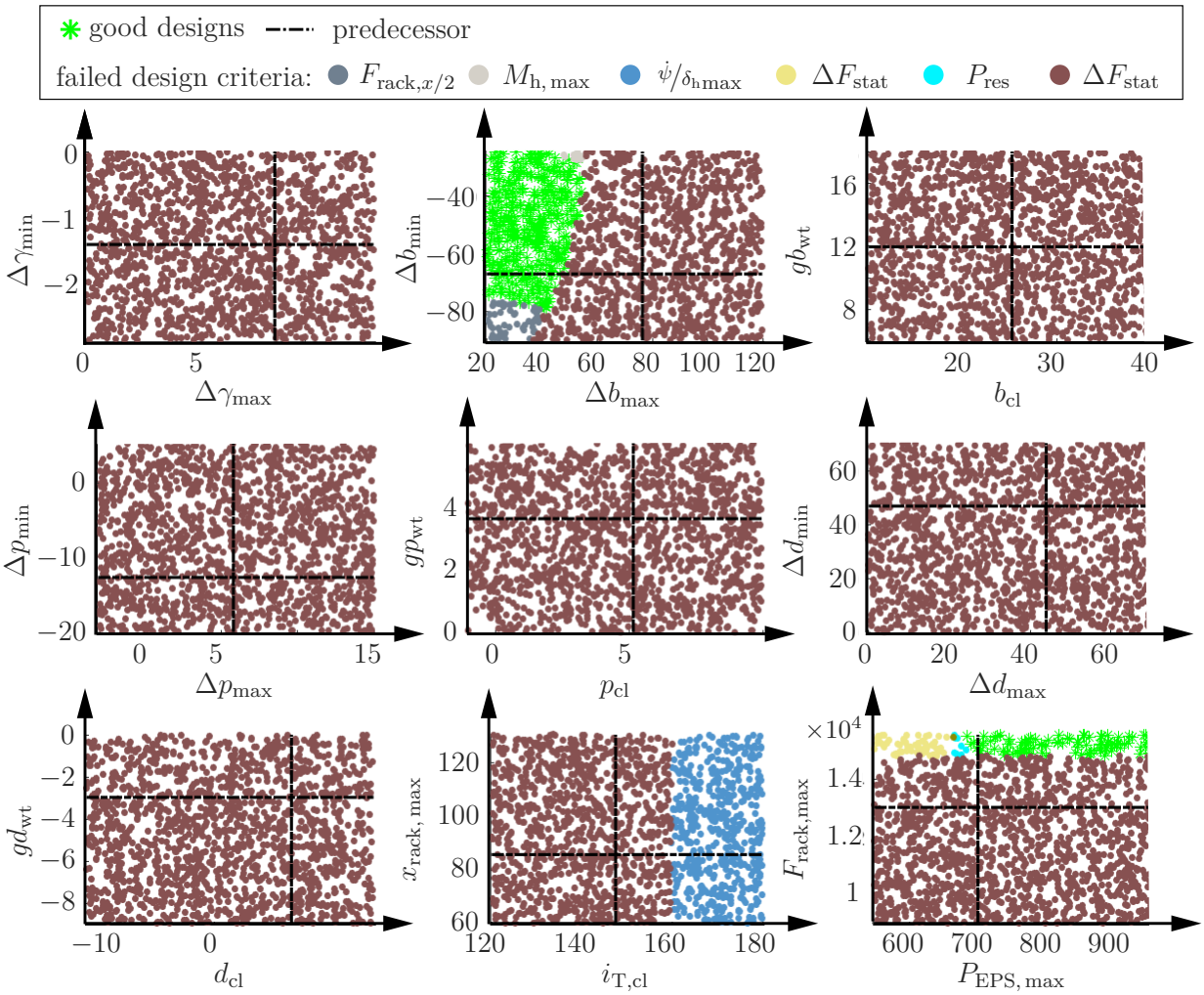


Figure 6.1: Initial section views of the design space for design scenario 1.

6.1 Design Objectives

An overview of the objective criteria considered for the following design problem is given in the Tables 6.1 and 6.2. The tables also provide information about the graph in which the CV is determined and whether the associated threshold value is an upper or lower bound or if both exist.

Table 6.1: Objective vehicle targets – part 1.

Category		Description	Symbol	lb	ub
self-steering behavior	QSSC	steady-state steering angle characteristics $\Delta\delta_{\mathrm{h},95\%} = \delta_{\mathrm{h}}\left(7\,\mathrm{m/s^2}\right) - \delta_{\mathrm{h}}\left(0.95\,a_{\mathrm{y,\,max}}\right)$	$\Delta\delta_{\mathrm{h}}\left(7\,\mathrm{m/s^2}\right)$, $\Delta\delta_{\mathrm{h},95\%}$	x	x
		steady-state body slip angle characteristics	$\Delta\beta\left(7\,\mathrm{m/s^2}\right)$	x	
		understeer gradient	$K\left(4\,\mathrm{m/s^2}\right)$	x	x
		maximum yaw velocity response and at $v = 190\,\mathrm{km/h}$, WEAVE	$\dot{\psi}/_{\delta_{\mathrm{h}}\mathrm{max}}$, $\dot{\psi}/_{\delta_{\mathrm{h}}190}$	x	x
		time delay between steering input and yaw velocity, CSST	T_{eq}		x
limit behavior	QSSC	wheel load while cornering	$F_{\mathrm{z,\,min}}$	x	
		maximum lateral acceleration	$a_{\mathrm{y,\,max}}$	x	
steering torque	QSSC	maximum steering torque	$M_{\mathrm{h,\,max}}$	x	x
		steering torque at maximum lateral acceleration	$M_{\mathrm{h}}\left(a_{\mathrm{y,\,max}}\right)$	x	x
		lateral acceleration at maximum steering torque	$a_{\mathrm{y}}\left(M_{\mathrm{h,\,max}}\right)$	x	x
		steering torque gradient at $a_{\mathrm{y}} = 1\,\mathrm{m/s^2}$	$gM_{\mathrm{h}}\left(1\,\mathrm{m/s^2}\right)$	x	x
		steering torque loss	$\Delta M_{\mathrm{h,loss}}$	x	x
	RAST	steering torque demand at $a_{\mathrm{y}} = 1\,\mathrm{m/s^2}$	$M_{\mathrm{h}}\left(1\,\mathrm{m/s^2}\right)$		x
turning circle	analytical	maximum wheel steer angle of the inner and outer wheel	$\delta_{\mathrm{f,inner,\,max}}$, $\delta_{\mathrm{f,outer,\,max}}$	x	x
		Ackermann angle error	$\delta_{\mathrm{AM,err}}$	x	x
		minimum turning radius	$R_{\mathrm{track,min}}$		x

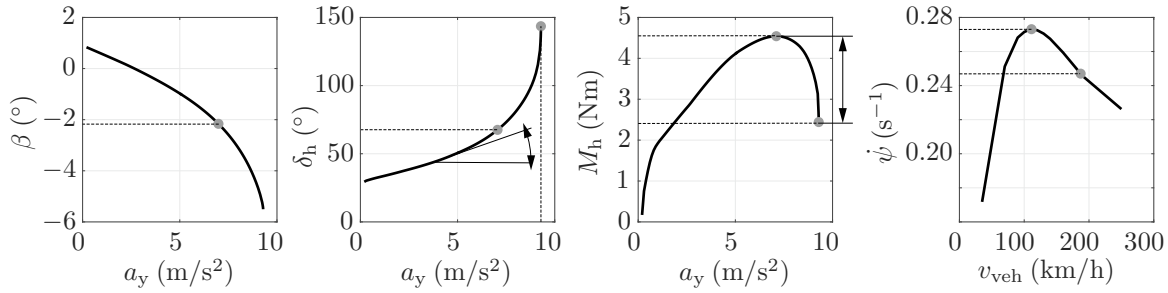
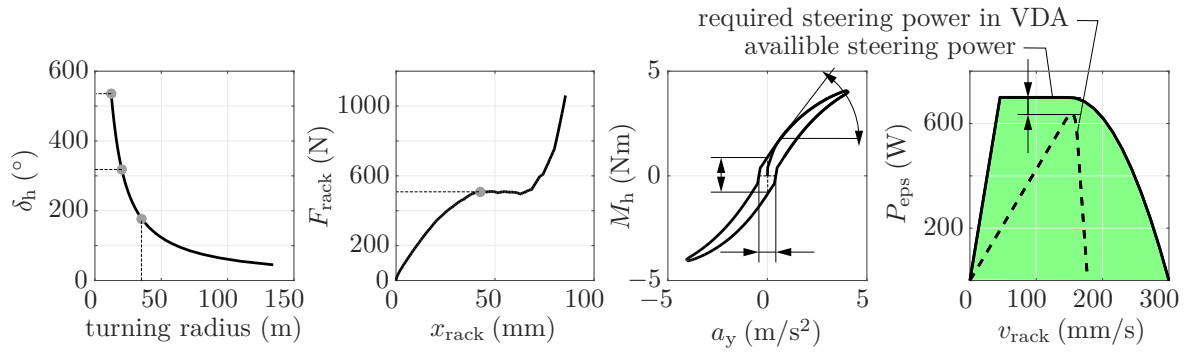


Table 6.2: Objective vehicle targets – part 2.

Category		Description	Symbol	lb	ub
parking behavior	RAST	turning radius at $\delta_\mathrm{h} = 180°$	$R_\mathrm{track}\,(\delta_\mathrm{h} = 180°)$		x
		steering rack force at half steering rack travel	$F_{\mathrm{rack},x/2}$	x	
		maximum steering wheel angle	$\delta_{\mathrm{h,\,max}}$		x
		steering angle for a turning radius of 15 m	$\delta_{\mathrm{h,15\,m}}$		x
directional stability	WEAVE	vertical deflection of the vehicle body on the front and rear axle	$z_\mathrm{FA,jacking}$, $z_\mathrm{RA,jacking}$		x
		vertical and horizontal hysteresis of M_h over a_y	$M_\mathrm{h,hyst}$, $a_\mathrm{y,hyst}$		x
		gradient of the steering torque over lateral acceleration in weave at $a_\mathrm{y} = 1\,\mathrm{m/s^2}$	$gM_\mathrm{h,hyst}$	x	x
steering power	SWD	dynamic steering torque	$M_\mathrm{h,dyn}$		x
		steering power deficit	P_deficit	x	
		steering power at k percent of steering rack stroke ($k = 15, 30, 60, 95$)	$P_{k\%}$	x	
	parking	difference between required and available steering power dynamically and stationary	ΔP_dyn, ΔP_stat	x	
		difference between occurring and permissible steering rack force	ΔF_stat	x	
other		maximum yaw rate while the passage of wheel ruts	$\dot{\psi}_{\mathrm{ruts,\,max}}$		x
		steering torque reserve	$M_\mathrm{h,res}$	x	x
		steering power reserve	P_res	x	x



The objectified overall vehicle targets are the main input of the method. They are directly related to the subjective evaluation of the vehicle behavior by the customer. To measure those objectified vehicle characteristics the following maneuvers must be virtually driven:

- quasi-steady-state cornering (QSSC), ISO 4138 (2012)
- ramp steer (RAST), ISO 4138 (2012)
- weave test (WEAVE), ISO 13674-1 (2003)
- continuous sine steer (CSST), ISO 7401 (2003)
- sine with Dwell (SWD), ISO 19365 (2016)

6.2 Design Variables

The design variables are clustered hierarchically into two groups: one group that represents the subsystem performance (y-level) and another that includes the design variables of the subsystems (x-level). In the following, the characteristic values of both are described.

6.2.1 Design Variables of the System

The design variables of the system are basically characteristic values of the different subsystems and provide the requirements for the detail design of the associated subsystems. Table 6.3 presents an overview of the characteristic values for the design of the steering system considered in this example. Since a smaller EPS-motor should be used for the vehicle, its maximum power and adjustable force are limited, therefore both design variables may not be shifted. However, the maximum rack stroke, and also pinion ratio, may be adjusted in order to satisfy the requirements.

Table 6.3: Design variables of the system belonging to the steering subsystem.

Name	Symbol
max. rack stroke (mm)	$x_{\text{rack, max}}$
max. power EPS (W)	$P_{\text{EPS, max}}$
max. force steering rack (N)	$F_{\text{rack, max}}$
pinion ratio (mm/rev)	i_{c}

The design variables for vehicle design that are associated with the suspension system are presented in Table 6.4. Since a kinematic model is used for the suspension design, the feasibility of some characteristic values may not be evaluated by it. Therefore, the fields with a gray background indicate characteristic values on which no requirements are derived top-down using the vehicle simulation model but by the engineer based on experience. As a result, satisfying both the vehicle requirements and feasibility restrictions may be considered simultaneously. Empty fields indicate that no such characteristic value exists, as the particular design variable is not sensitive to that type of variation.

Table 6.4: Design variables of the system belonging to the suspension subsystem.

Name	Symbol	Value at construction level	Gradient vs. wheel travel (/100mm)	Variation until min. steering angle	Variation until max. steering angle	Gradient vs. steering angle (/°)	Gradient vs. lateral force (/N)
toe (min)	δ_t	$\delta_{\mathrm{t,cl}}$	$g\delta_{\mathrm{t,wt}}$				$g\delta_{\mathrm{t,F_y}}$
camber (°)	γ	γ_cl	$g\gamma_\mathrm{wt}$	$\Delta\gamma_\mathrm{min}$	$\Delta\gamma_\mathrm{max}$		$g\gamma_{\mathrm{F_y}}$
scrub radius (mm)	d	d_cl	gd_wt	Δd_min	Δd_max		
kinematic trail (mm)	b	b_cl	gb_wt	Δb_min	Δb_max	gb_δ	
steering linkage transmission ratio (mm/°)	i_T	$i_{\mathrm{T,cl}}$		$\Delta i_{\mathrm{T,\,min}}$	$\Delta i_{\mathrm{T,\,max}}$		
wheel-load lever (mm)	p	p_cl	gp_wt	Δp_min	Δp_max	gp_δ	
roll center height (mm)	RC_h	$RC_{\mathrm{h,cl}}$	$gRC_{\mathrm{h,wt}}$				

6.2.2 Design Variables of the Subsystems

The design variables of the subsystems belong to the detail level (x-level) and must be adapted in order to fulfill the requirements on the subsystems. In the following, they are divided into groups that belong to a certain subsystem.

Design Variables of the Suspension System

The design variables of the suspension system are the positions of the kinematic joints and the stiffnesses of the bushings. In this example, only the kinematic joint positions ($\boldsymbol{a}$,$\boldsymbol{b}$,$\boldsymbol{c}$,$\boldsymbol{d}$,$\boldsymbol{e}$,$\boldsymbol{f}$,$\boldsymbol{g}$,$\boldsymbol{h}$,$\boldsymbol{i}$, see Figure 4.3) are considered to be adopted in all three directions, since only a kinematic model is available for the suspension. Therefore, twenty-seven design variables need to be adapted in order to fulfill the requirements on the suspension system. For the design and analysis of an elasto-kinematic suspension model using solution spaces see (Wimmler et al. 2016), where the stiffnesses of measured bushings were scaled in all three directions.

Design Variables of the Steering System

For the design of the steering system, no quantitative model is available. Therefore, the feasibility is estimated using the data in Figure 2.10 and needs to be double checked by the EPS supplier. Since the concept of the steering system impacts the required space, a close cooperation between functional and geometrical integration is necessary.

6.3 Underlying Quantitative Design Models

The foundation for the following design procedures is built by two simulation models in form of neural networks, one for the vehicle design and one for the suspension design.

6.3.1 Neural Network Model for Vehicle Design

The neural network of the vehicle model is a one layer feedforward network and consists of 67 degrees of freedom and 50 objective criteria (see Table 6.5). The optimization algorithm for the creation of the neural network determined that, for a particular objective criterion, 24 neurons are necessary in order to represent the quantitative relations between design variables and objective criterion with sufficient accuracy.

Table 6.5: Properties of the neural network set for vehicle design.

one layer feedforward network	
degrees of freedom:	67
number of objective criteria:	50
max number of neurons:	24

6.3.2 Neural Network Model for Suspension Design

As a quantitative model for the suspension system, a one layer feedforward network is used as well. The properties of the applied neural network are presented in Table 6.6. Since the causal relations between design variables and objective criteria are less complex compared to those of the vehicle model, the artificial suspension model needs to handle a smaller number of neurons in order to obtain accurate fitting results.

Table 6.6: Properties of the neural network set for suspension design.

one layer feedforward network	
degrees of freedom:	31
number of objective criteria:	32
max number of neurons:	19

6.4 Evaluation of the Design for Vehicle Dynamics

The fully-automated algorithm presented in Section 5.2 is applied to the *Design Scenario 1.* Based on the objective criteria of the overall vehicle presented in Section 6.1, the algorithm is used to identify a consistent set of subsystem requirements as well as a proposal for a robust design on a detail level. Therefore, the following three stages are applied repeatedly in an iterative optimization scheme:

- Stage 1: Searching for a design on the subsystem level (y-level) that satisfies all requirements on the overall vehicle targets by *robust optimization.*
- Stage 2: Enlarging the particular robust design point identified in stage 1 to a solution box.

- Stage 3: Finding a feasible design on the detail level (x-level) that satisfies the requirements on the subsystem level (y-level) defined in stage 2.

Convergence of the overall optimization procedure is achieved by the exchange of information between the three stages. In the following, the first iteration loop is reviewed in detail before the final results are presented.

Stage 1: robust optimization. In stage 1 of the algorithm, a robust optimization (see Subsection 2.3.3) is applied to the vehicle. The optimization goal is to find a single design on the subsystem level (y-level) that first satisfies all requirements on the overall vehicle performance and second, provides a maximized robustness. Robustness, in this case, is measured within the design space. A robust design is optimized to have a large distance to the closest bad design in all dimensions.

For the underlying design scenario, a good design satisfying all targets on the vehicle is found by use of the performance function described in Equation (5.12). Since no information about the position of the closest feasible design of the suspension are available within the first iteration loop in this particular example, the robustness is evaluated by the following equation:

$$\varphi_{\mathrm{rb}} = -\left(\frac{\Delta b^{\mathrm{lb}}_{\mathrm{rob,cl}}\, \Delta b^{\mathrm{ub}}_{\mathrm{rob,cl}}}{b^{\mathrm{ub}}_{\mathrm{ds,cl}} - b^{\mathrm{lb}}_{\mathrm{ds,cl}}} \; \frac{\Delta g b^{\mathrm{lb}}_{\mathrm{rob,wt}}\, \Delta g b^{\mathrm{ub}}_{\mathrm{rob,wt}}}{g b^{\mathrm{ub}}_{\mathrm{ds,wt}} - g b^{\mathrm{lb}}_{\mathrm{ds,wt}}} \; \frac{\Delta g b^{\mathrm{lb}}_{\mathrm{rob},\delta}\, \Delta g b^{\mathrm{ub}}_{\mathrm{rob},\delta}}{g b^{\mathrm{ub}}_{\mathrm{ds},\delta} - g b^{\mathrm{lb}}_{\mathrm{ds},\delta}} \right.$$
$$\left. \frac{\Delta\Delta b^{\mathrm{lb}}_{\mathrm{rob,\,min}}\, \Delta\Delta b^{\mathrm{ub}}_{\mathrm{rob,\,min}}}{\Delta b^{\mathrm{ub}}_{\mathrm{ds,\,min}} - \Delta b^{\mathrm{lb}}_{\mathrm{ds,\,min}}} \; \frac{\Delta\Delta b^{\mathrm{lb}}_{\mathrm{rob,\,max}}\, \Delta\Delta b^{\mathrm{ub}}_{\mathrm{rob,\,max}}}{\Delta b^{\mathrm{ub}}_{\mathrm{ds,\,max}} - \Delta b^{\mathrm{lb}}_{\mathrm{ds,\,max}}} \; \dots \right) \tag{6.1}$$

Consider that:

- Weighting factors are equal for each design variable.
- b indicates a parameter related to the kinematic trail of the suspension.
- *lb/ub* indicates whether the lower bound or the upper bound of the parameter is meant.
- *rob* indicates the distance between the current design and the associated next bad design, if only this parameter is varied.

Therefore, the result of the first iteration loop is partially presented in Figure 6.2. The result shows that a robust design within the solution space could be found, since the next bad design is far away from the design found by the optimization. Nevertheless, the subsystem performance of the predecessor's suspension does not lie within the solution space as it becomes clear by viewing the steering linkage transmission ratio at construction level $i_{\mathrm{T,cl}}$. The following two stages of the optimization algorithm (stage 2 and stage 3) will point out whether this robust design needs to be adapted in order to enable feasibility or not.

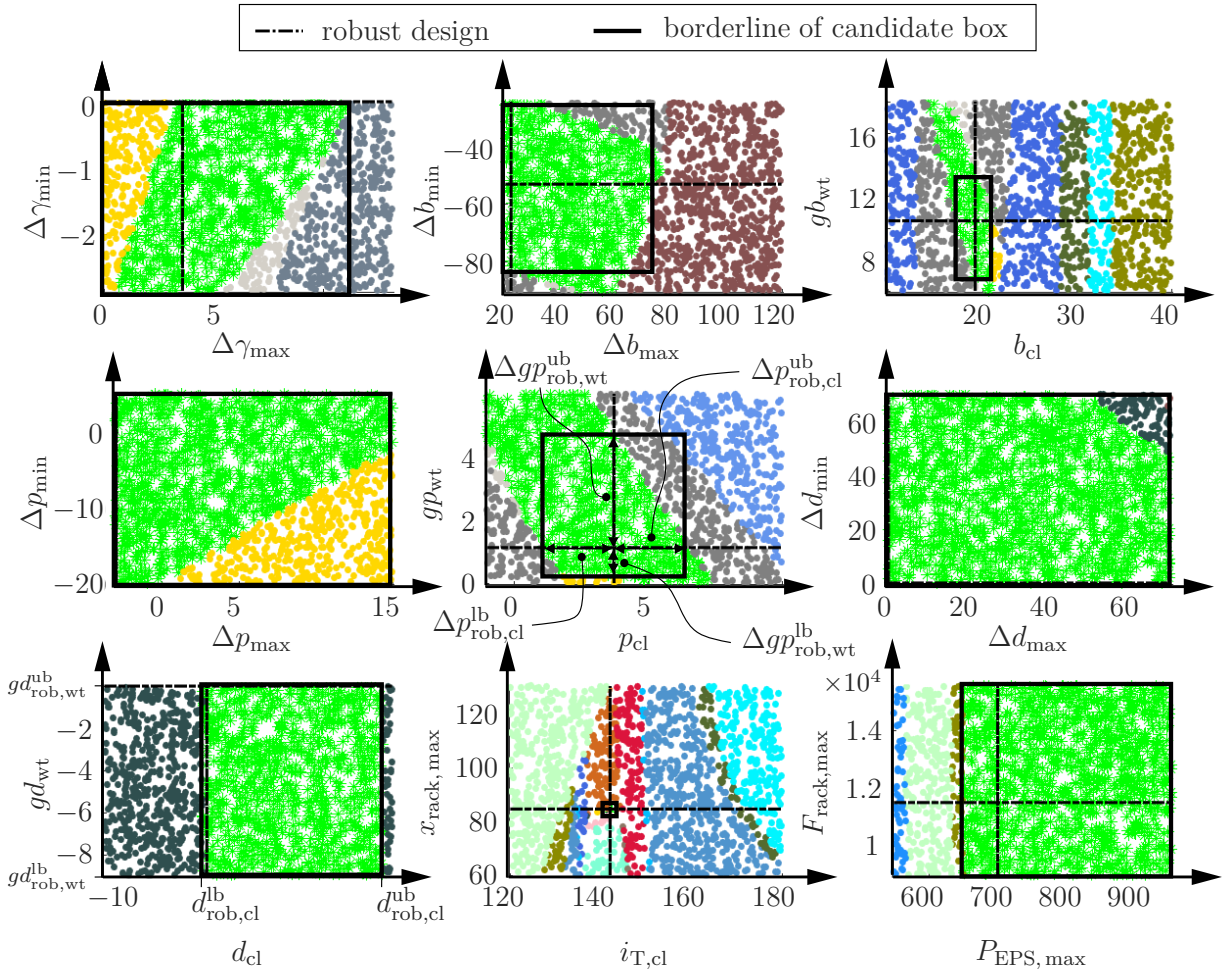


Figure 6.2: Section view of the design space after the first robust design optimization.

Stage 2: Solution Box Optimization. In stage two, the robust design following stage 1 is expanded to a solution box as described in Subsection 2.3.4. Since no information about feasibility is available yet, the system performance φ_{sb} is equal to the negative box size measure $-\mu_{\mathrm{sb}}$. If the permitted percentage of bad designs exceeds the specified critical value, a penalty equal to $\mu_{\mathrm{sb}}\,(B_{\%} - B_{\%,\mathrm{crit}})$ is added to the box size measure μ_{sb}. In order to increase the possibility of finding a feasible design on the detail level (x-level), the size measure μ_{sb} of the solution box $\Omega^{\mathrm{y}}_{\mathrm{sb}}$ is maximized $\mu(\Omega) \rightarrow \max$. According to Equation (5.14), which represents the general case, the vehicle/system performance is calculated by

$$\varphi_{\mathrm{sb}} = \left(\frac{b^{\mathrm{ub}}_{\mathrm{cl}} - b^{\mathrm{lb}}_{\mathrm{cl}}}{b^{\mathrm{ub}}_{\mathrm{ds,cl}} - b^{\mathrm{lb}}_{\mathrm{ds,cl}}} \; \frac{gb^{\mathrm{ub}}_{\mathrm{wt}} - gb^{\mathrm{lb}}_{\mathrm{wt}}}{gb^{\mathrm{ub}}_{\mathrm{ds,wt}} - gb^{\mathrm{lb}}_{\mathrm{ds,wt}}} \; \frac{gb^{\mathrm{ub}}_{\delta} - gb^{\mathrm{lb}}_{\delta}}{gb^{\mathrm{ub}}_{\mathrm{ds},\delta} - gb^{\mathrm{lb}}_{\mathrm{ds},\delta}} \right. \\ \left. \frac{\Delta b^{\mathrm{ub}}_{\mathrm{min}} - \Delta b^{\mathrm{lb}}_{\mathrm{min}}}{\Delta b^{\mathrm{ub}}_{\mathrm{ds,\,min}} - \Delta b^{\mathrm{lb}}_{\mathrm{ds,\,min}}} \; \frac{\Delta b^{\mathrm{ub}}_{\mathrm{max}} - \Delta b^{\mathrm{lb}}_{\mathrm{max}}}{\Delta b^{\mathrm{ub}}_{\mathrm{ds,\,max}} - \Delta b^{\mathrm{lb}}_{\mathrm{ds,\,max}}} \; \dots \right). \tag{6.2}$$

in this particular case. Starting from the robust design following stage 2, the upper bounds are only able to increase, while the lower bounds may only decrease. Therefore, the result of the solution box optimization is presented in Figure 6.3.

Stage 3: Feasibility Optimization. The solution box found by the algorithm serves as input for the feasibility optimization in stage 3. Since the position and size of the solution box changes in each iteration, the existence of a feasible suspension satisfying all requirements derived on it must be proven. Therefore, the position of the kinematic joints is adapted in order to first, satisfy all requirements, and second, to find a design that lies in the center of the target region. Therefore, the performance function presented in Equation (4.3) for the general case is used.

In stage 3 of iteration 1, the performance of the first four out of 25 objective criteria is calculated by Equation (6.3). Since the performance values of $\Delta\gamma_{\max}$ and $\Delta\gamma_{\min}$ are not contained within the target intervals, a penalty is added regarding their distance to the target interval.

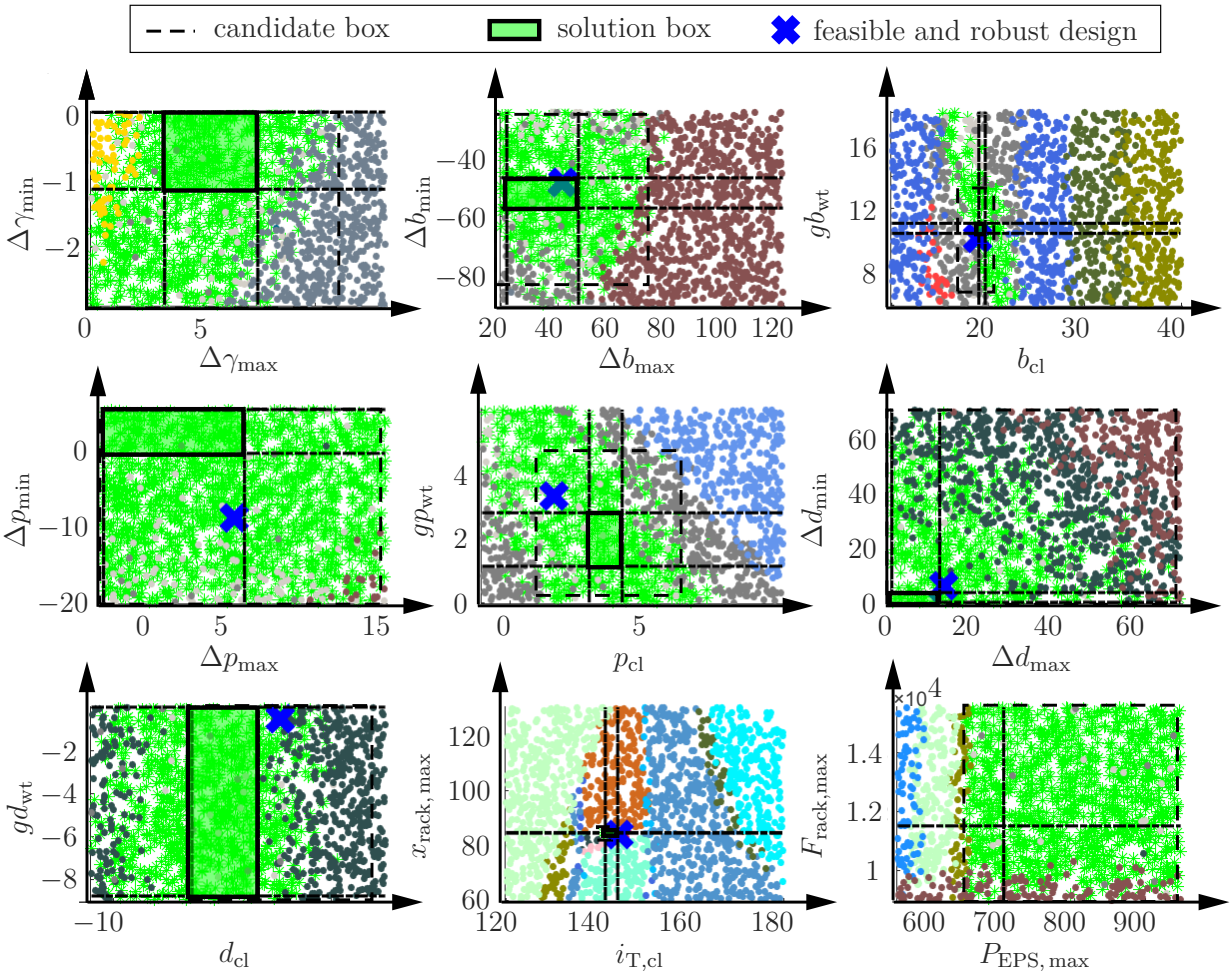


Figure 6.3: Results of the first iteration loop.

$$\varphi_{\text{feas}} = -25/2 \left(\left| \frac{\Delta\gamma_{\max} - \Delta\gamma_{\max}^{\text{target}}}{\Delta\gamma_{\max}^{\text{ub}} - \Delta\gamma_{\max}^{\text{lb}}} \right| + \frac{\min(|\Delta\gamma_{\max} - \Delta\gamma_{\max}^{\text{lb}}|,\ |\Delta\gamma_{\max}^{\text{ub}} - \Delta\gamma_{\max}|)}{\Delta\gamma_{\max}^{\text{ub}} - \Delta\gamma_{\max}^{\text{lb}}}\, p + \left| \frac{\Delta\gamma_{\min} - \Delta\gamma_{\min}^{\text{target}}}{\Delta\gamma_{\min}^{\text{ub}} - \Delta\gamma_{\min}^{\text{lb}}} \right| + \frac{\min(|\Delta\gamma_{\min} - \Delta\gamma_{\min}^{\text{lb}}|,\ |\Delta\gamma_{\min}^{\text{ub}} - \Delta\gamma_{\min}|)}{\Delta\gamma_{\min}^{\text{ub}} - \Delta\gamma_{\min}^{\text{lb}}}\, p + \left| \frac{\Delta b_{\min} - \Delta b_{\min}^{\text{target}}}{\Delta b_{\min}^{\text{ub}} - \Delta b_{\min}^{\text{lb}}} \right| + \left| \frac{\Delta b_{\max} - \Delta b_{\max}^{\text{target}}}{\Delta b_{\max}^{\text{ub}} - \Delta b_{\max}^{\text{lb}}} \right| \dots \right) \tag{6.3}$$

Figure 6.3 presents the results of the first iteration loop in a vivid manner. Therefore, the candidate box from stage 1, the solution box from stage 2, and the most likely feasible design resulting from stage 3 are printed. Suppose that each subfigure presents a projection of the permissible intervals of the not shown 25 subsystem properties as declared in Subsection 2.3.5. As the performance of the feasible design found on the detail level does not lie within the solution space, feasibility could not be established in this case. Therefore, additional iteration loops are necessary in order to rebalance the permissible intervals of the solution box.

Final Results. The final results show that feasible permissible intervals could be derived on subsystems in a top-down process (see Figure 6.4) in order to satisfy the overall vehicle targets. Figure 6.5 presents the results of the automatic optimization algorithm and proves that a consistent design satisfying all overall vehicle targets and including a greatly reduced maximum steering rack force could be found. Although the final solution box on the subsystem level (y-level) is smaller compared with the biggest found within the first iteration loop of the algorithm, it is still the preferred one. The shift towards a smaller solution box established the feasibility of the requirements derived on the suspension-subsystem, which is essential for a consistent design.

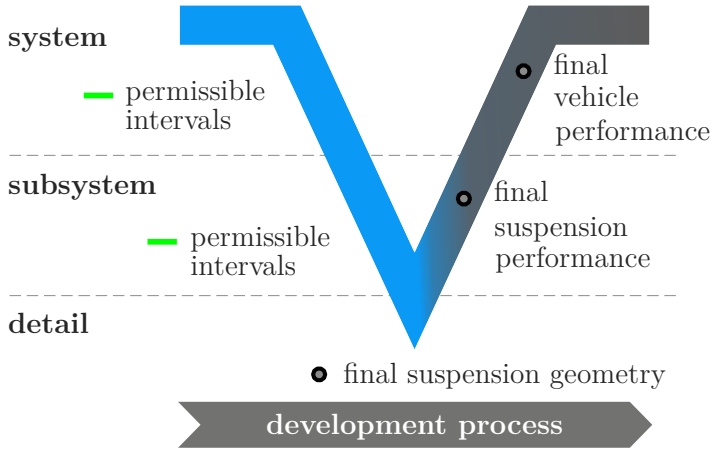


Figure 6.4: Development process using solution spaces and target cascading.

Flexibility and Robustness. The final solution box offers great flexibility for the kinematic joint positions, since its size measure is maximized with respect to the feasibility of their permissible intervals. Therefore, the positions of the kinematic joints may be varied as long as the suspension performance lies within those target regions. Robustness could be considered by first, maximizing the size of the solution box and second, finding a setup of kinematic joint positions close to the center of the target regions. Both properties of the result increase the distance to system failure on the y-level.

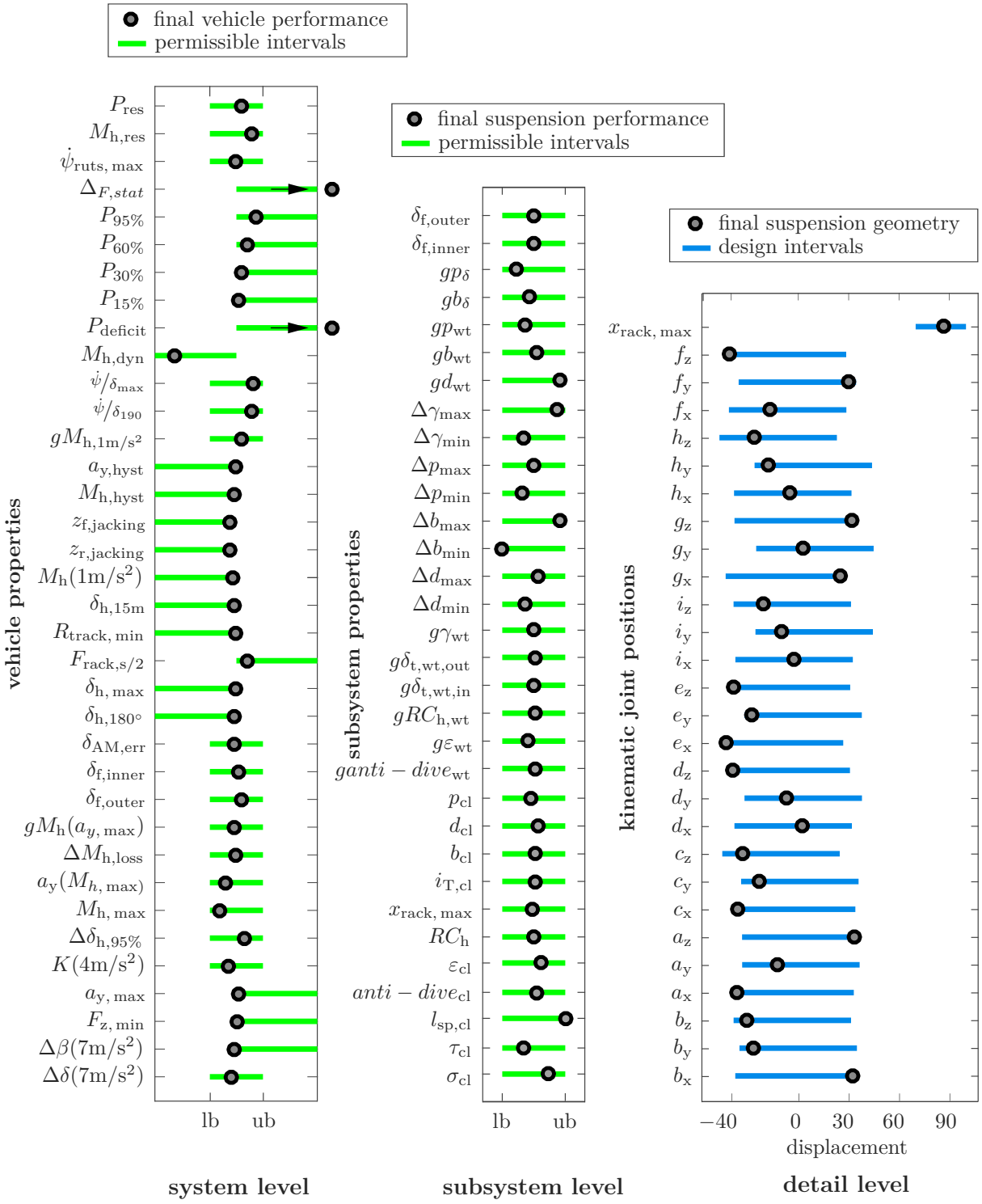


Figure 6.5: Final results of the automatic algorithm.

After the final suspension geometry is defined, it can be integrated in the CAD-environment by a table including the absolute kinematic joint positions, see Figure 6.6. In the following design process, each component is designed more in detail.

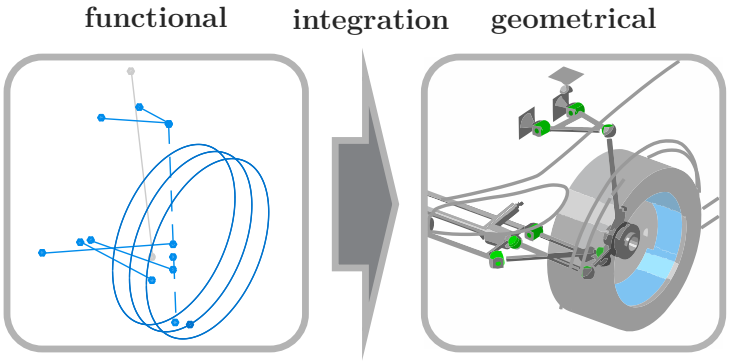


Figure 6.6: Automatic integration of the results into the CAD-environment.

6.5 Evaluation of the Design for Vehicle Platforms

Design Scenario 2. Suppose that two vehicles of the same vehicle platform are designed. Both vehicles are treated as separate systems "s". One vehicle is a full-size luxury class vehicle s_{LC}, and the other one is an upper-class vehicle s_{UC}. Therefore the vehicle-DNA of each vehicle is different.

The luxury class vehicle is a 4-door sedan. Consequently, a safe and comfortable behavior is expected by the customers. The upper-class vehicle is a gran turismo, therefore its character needs to be sportier to satisfy customer expectations. To ensure the intended behavior of the vehicles, the same 36 objective criteria as in design scenario 1 are considered to measure the vehicle performance. Consequently, different permissible intervals are prescribed on the system level for each vehicle to differentiate their character. In Figure 6.7, only the permissible intervals of those design criteria that primarily limit the expansion of the solution space in the final iteration of the optimization are presented. Under other conditions (e.g. iteration steps, design objectives, geometrical constraints) other design criteria may be more important. If all 36 permissible intervals of the objective criteria contain the vehicle performance, the vehicle is good. Three dots at the lower or upper bound of a permissible interval in Figure 6.7 indicate an open interval. In that particular case, only one threshold value exists.

The rear suspension of both vehicles should not differ from the predecessor. Therefore, no changes are allowed. However, the front suspension of both vehicles will be new such that it can be adapted with respect to the design objectives under consideration of the geometrical and feasibility constraints. In addition, both vehicles are intended to share the same front axle subframe. Tires are permitted to be different for each vehicle as long as the front and rear tires of one vehicle are the same. The tire diameter of both vehicles is the same and also predefined. As steering gear, the same existing EPS-APA system is used for both vehicles. However, the maximum rack stroke may be adapted. To satisfy the top-level design objectives, 31 permissible intervals may be adapted on the subsystem level by the numerical optimization for each vehicle separately. Therefore, by considering the upper and lower bound of a permissible interval as separate design variables, the total number of design variables on the subsystem level is 124.

For both vehicles, the resulting final solution box is presented in Figure 6.8. To create those results, eight iterations of the automatic algorithm presented in Section 5.2 were necessary. As the feasibility optimization was used to derive feasible requirements on the suspension system, the resulting subsystem performance is marked within the solution box. The feasible design of s_{LC} (dots) is contained in the solution box of s_{LC} (solid line) and the feasible design of s_{UC} (crosses) is contained in the solution box of s_{UC} (dashed line). Therefore, the requirements on both suspension systems are feasible. Feasibility of the requirements on the tires is ensured by the method introduced in Chapter 3 using SVM. Based on the example of ${}_{LC}K_{\alpha}$, ${}_{LC}\mu_{\mathrm{y,max}}$, ${}_{LC}K_{\mathrm{Mz}}$ and ${}_{LC}M_{\mathrm{z,max}}$, Figure 6.9 proves that the method converged towards a feasible solution by presenting projections of the solution space. Areas without sample points indicate the regions where no tire can be built. The permissible intervals of the FTCs may also be represented by areas of characteristic curves that lie in the solution space and are feasible, see Figure 6.10. The stiffnesses of the anti-roll bars c_{ARB} do not need to be examined for feasibility as they can be realized easily through the adaption of their diameter or wall thickness.

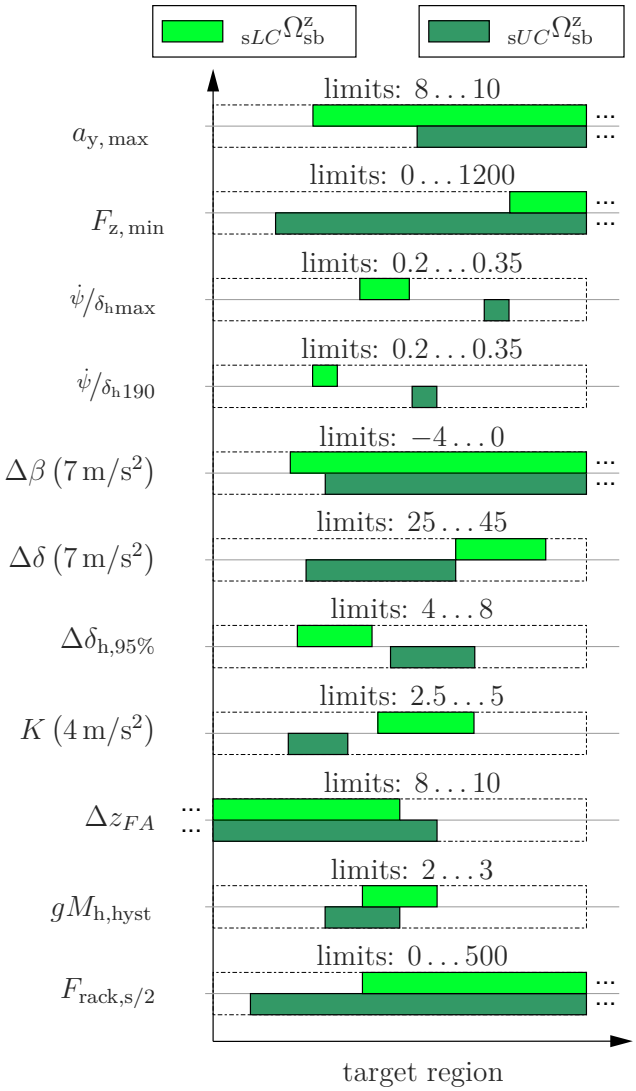


Figure 6.7: Differences in the permissible intervals of the objective criteria on the z-level. The rest of the 36 objective criteria that were considered in this design scenario are presented in Figure 6.5.

Interpretation of the results on the subsystem level. Compared to the sedan, the gran turismo s_{UC} needs tires with an increased lateral grip coefficient $\mu_{\mathrm{y,max}}$ in order to achieve the required lateral acceleration. A shorter relaxation length σ_{y0} of the tire results in less time delay between steering input and yaw velocity T_{eq} which improves the dynamic behavior of the vehicle. The yaw velocity response ($\dot{\psi}/\delta_{\mathrm{hmax}}$ and $\dot{\psi}/\delta_{\mathrm{h}190}$) in this design problem is mainly adjusted by the steering linkage transmission ratio at construction level $i_{\mathrm{T,cl}}$. Due to the negative correlation between both $\dot{\psi}/\delta_{\mathrm{h}}$ and $i_{\mathrm{T,cl}}$, the $i_{\mathrm{T,cl}}$ is lower for the gran turismo in order to achieve a more direct self-steering behavior. As presented by the rhombi in Figure 6.9, the solution box of the sedan ${}_{sLC}\Omega^{y}_{sb}$ contains no existing

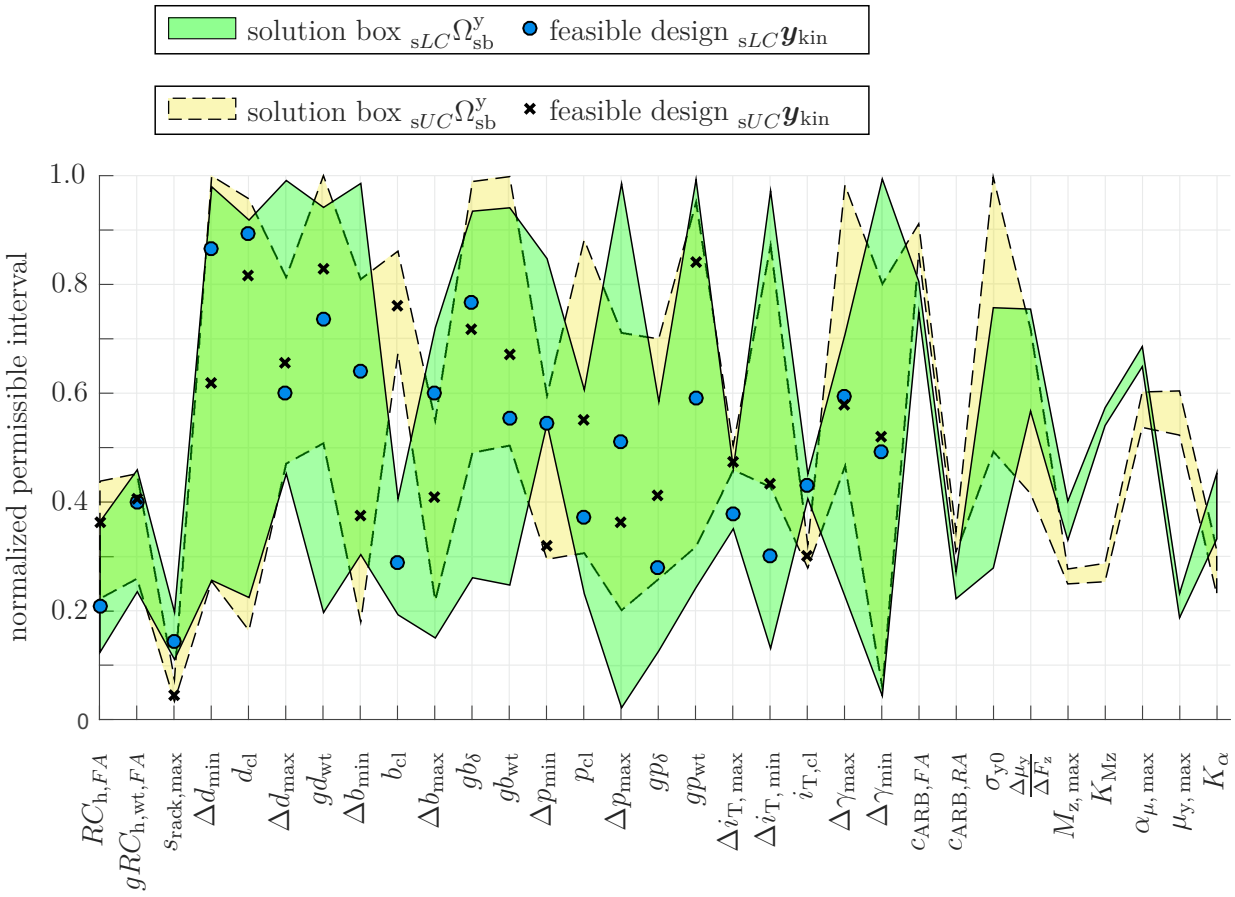


Figure 6.8: MPP showing the solution boxes for both vehicles on the subsystem level normalized to the design space. The feasible subsystem performance of the suspension systems is also marked.

tire. Existing tires are only plotted in Figure 6.9 if $\alpha_{\mu,\max}$, $\frac{\Delta\mu_y}{\Delta F_z}$ and σ_{y0} are contained in their associated permissible intervals. As a result, the engineer may adapt the position of the solution box in order to use an existing tire, by help of the semi-automatic algorithm presented in Section 5.3. However, it is also possible to develop a new tire that satisfies all permissible intervals on the FTCs. By presenting the solution box of the FTCs through areas of characteristic curves, it can be easily evaluated whether a measured tire satisfies the requirements by overlaying its characteristic curve. Another advantage is that the engineer is familiar with this visualization. However, Figure 6.10 (a) also shows a disadvantage. If the relation between the value ranges of both diagram axis and the permissible intervals is unfavorable, then the permissible intervals seem to be very small.

In Figure 6.11, the resulting suspension systems are shown. The blue suspension system represents the sedan s_{LC}, while the black one represents the suspension system of the gran turismo s_{UC}.

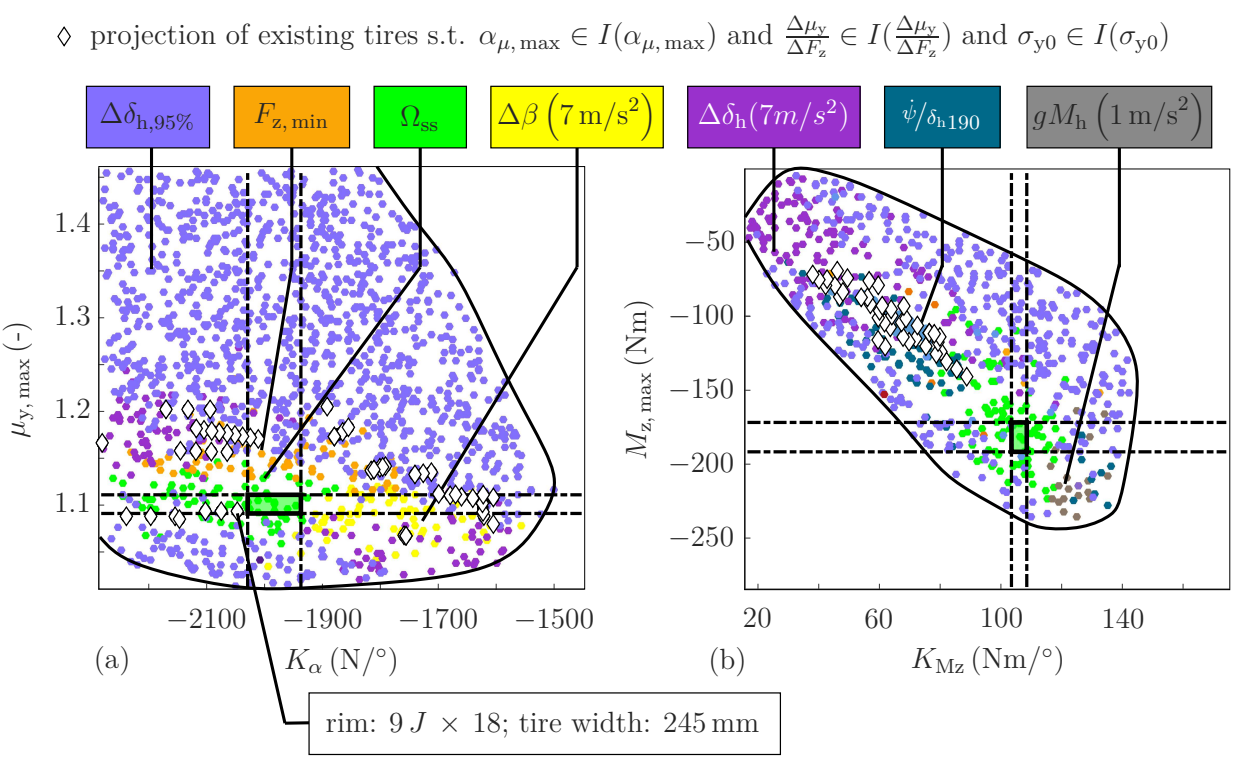


Figure 6.9: Feasibility of the requirements on the tire of the luxury class 4-door sedan visualized by projections through the solution space.

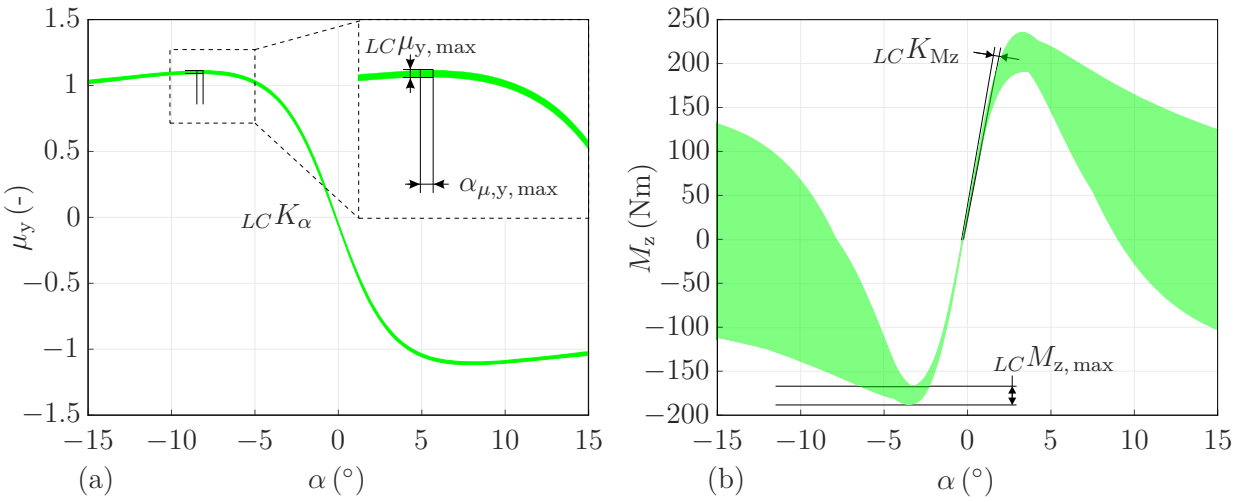


Figure 6.10: Permissible intervals of $_{LC}K_\alpha$, $_{LC}\mu_{y,\max}$, $_{LC}K_{Mz}$ and $_{LC}M_{z,\max}$ as areas for characteristic curves.

Interpretation of the results on the detail level. As an optimization result, the same front axle subframe can be used since the relative distances between the kinematic joint positions are identical between both systems and the subsystem requirements are satisfied. The kinematic trail of the gran turismo is larger at the construction level compared to the sedan, which can be seen in Figure 6.8 and 6.11. For both suspension systems, multiple

geometrical constraints need to be considered to avoid collisions between different parts. Geometrical constraints that directly reduce the design space are also presented by red rectangles in Figure 6.11. Therefore, the kinematic joint positions are not allowed to enter those rectangles such as:

- kinematic joint **b**, which is restricted towards the inside of the vehicle due to the length of the teeth on the steering rack and
- kinematic joint **i**, which is restricted towards the outside of the vehicle to avoid a collision with the brake disk.

In addition, the minimum or maximum distance between two different kinematic joints may be restricted which reduces the design space as well. For this design problem, the minimum distances between the following kinematic joints are restricted:

- **g** and **h** in order to provide enough space for the housings of both joints,
- **d** and **e** in order to provide enough space for the spring-damper system.

Context of the results. The design scenario 2 has proven that vehicle platforms for multiple vehicles can be derived over multiple hierarchical levels, under consideration of complex subsystems, with the help of solution spaces. By applying the automatic algorithm presented in Section 5.2, the two main goals were achieved:

- identifying a feasible solution box that satisfies all requirements, and
- integrating a common front axle subframe.

The closer the subsystem performance to the center of the solution box, the more robust the design on the detail against unintended variations. Therefore, the scrub radius at construction level is realized with less robustness for the sedan than for the gran turismo in the final design. The factors in the performance function may be adapted for adjusting the ratio between maximizing the size of the solution box on one hand and, on the other hand, shifting the solution box with respect to the feasible design on the detail level. However, it is also possible to finalize the result of the automatic algorithm by applying the semi-automatic algorithm. Therefore, the particular needs of a project regarding robustness and uncertainties can be considered. In general, the more components should be shared among different vehicles, the more difficult it is to find a robust solution. Therefore, without any common components, a more robust design could be realized for each vehicle. Additionally, the difference in the top-level vehicle targets and the provided vehicle-DNA is important when sharing components.

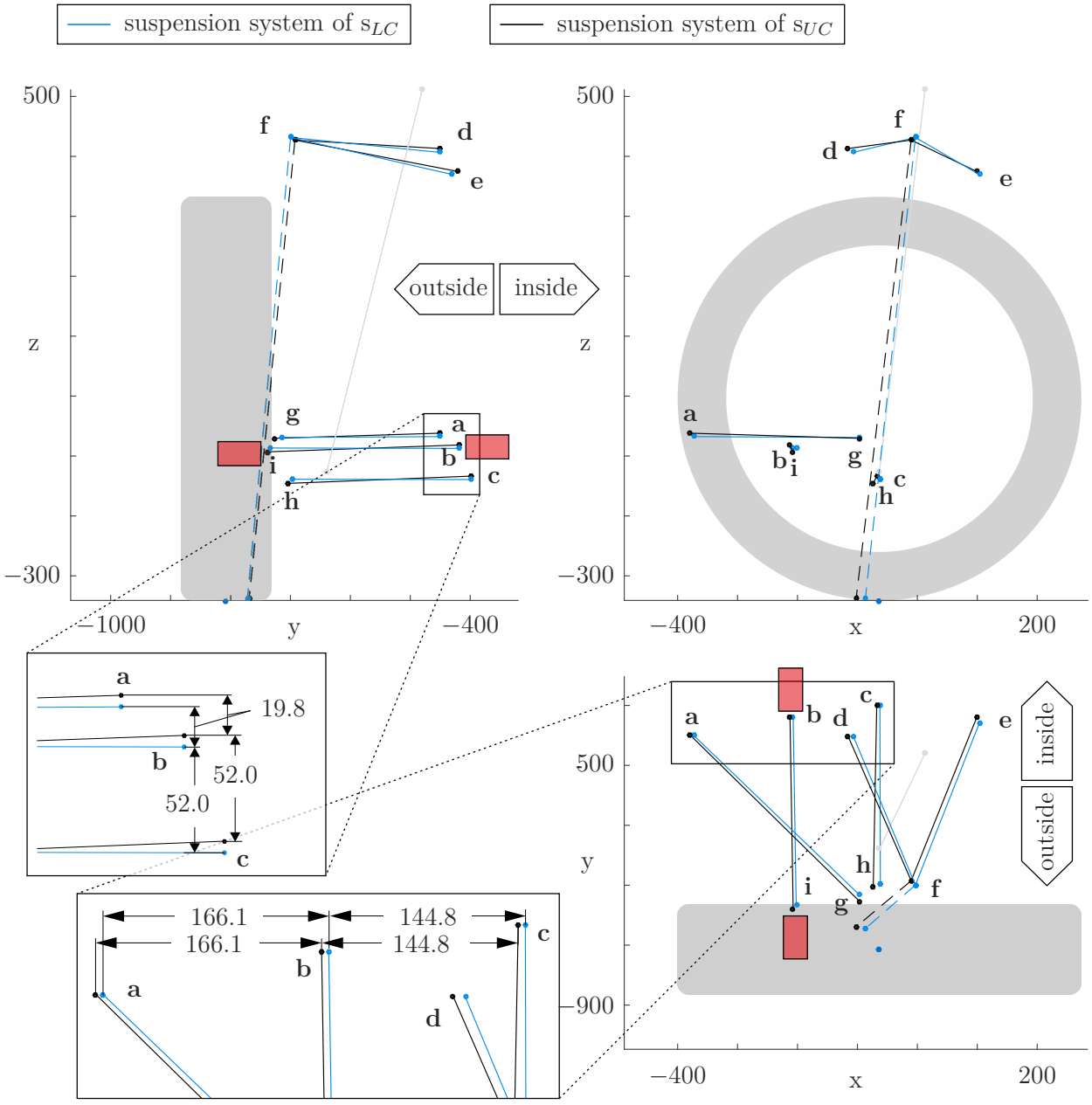


Figure 6.11: Suspension systems of two different vehicles that share the same front axle subframe.

CHAPTER 7

DISCUSSION

In the first chapter of this work, the need for an evolution of current design methods in the field of vehicle dynamics was pointed out. After reviewing the current state-of-the-art, action fields were identified, and three particular aims were formulated according to present and future challenges. Based on those aims, a solution approach was elaborated to master them.

The overall objective of the solution approach was a top-down optimization method that combines the benefits of target cascading and solution spaces plus allows the intuitive interaction with the user. To apply this concept to vehicle architectures and platforms, three subproblems concerning the following issues had to be solved first.

- Chapter 2: Architecture Design in Vehicle Dynamics.
- Chapter 3: Objectified Tire Development.
- Chapter 4: Integrated Design of Suspension Systems using Solution Spaces and Target Cascading.

In Chapter 5, the previously introduced methods were combined with each other to combine the benefits of target cascading and solution spaces. Therefore, an automatic and semi-automatic algorithm was developed. In Chapter 6, both algorithms were evaluated based on complex design problems.

Solution Spaces for Architecture Design

The architecture design regarding vehicle dynamics is challenging for many reasons, e.g., the large number of objective criteria and design variables as well as the complex relationships between them. This complexity is reflected in the large number of interdependencies

as well as their non-linearity. This work extends the scope of complexity that may be handled by a systems design approach. Therefore, it is the first that is able to simultaneously derive solution spaces for the vehicle-DNA, tires, suspension and the steering system.

Computational Effort

By using solution spaces for such complex design tasks, computational effort usually is an issue in our modern world with rapidly changing environmental conditions. The overall computational effort for obtaining an optimized solution box may be reduced mainly in two ways:

1. reducing the necessary computation time of a particular sample point and
2. reducing the number of necessary calculations to get a final result.

The necessary computation time of a particular sample point was greatly reduced by applying neural networks instead of physical quantitative models in the optimization algorithm. Since the neural network is created based on the physical model, its accuracy may be lower. However, the accuracy of the neural network created in this work is sufficient for the reasons described in Section A.1. To ensure a sufficient level of accuracy for such a complex system, a large number of sample points for training the neural network was necessary, as well as filter criteria for the performance measures, see Section A.3. There is a need for more advanced machine learning techniques in order to consider even more complex systems and to extend the considered design space.

The number of necessary calculations was reduced by the two-step optimization approach described in Section 5.2. Therefore, the starting point for the solution box optimization is shifted by a robust design optimization for each iteration of the outer loop, which is much less expensive than if the solution box needs to grow in a preferred direction. During the following solution box optimization, the starting point can grow in each direction. However, since the starting point always needs to be an element of the solution box, the possible regions for solution boxes as well as the computational effort are reduced. Current applications of the method have shown that there is a need to further reduce the computation time of the solution box optimization, as it represents the main part of the overall computation time. Therefore, further research regarding solution box optimization of complex systems is necessary.

Model Parameterization

Regarding the parameterization of the simulation model, several requirements were satisfied for enabling decentralized development of the subsystems. Therefore, parameterization was proposed, which not only ensures the fulfillment of the overall vehicle targets in terms of vehicle dynamics but also provides the different parties with flexibility in achieving the subsystem requirements.

For the tire behavior, a novel method was proposed in order to quantify the relationships between FTCs and objectified vehicle performance measures. Therefore, neural network

regression models and support vector machine classifiers were first combined to derive solution spaces. While the neural network evaluates the vehicle performance measures, the support vector machine divides the design space into feasible and non-feasible space. As the classifier assumes that all sets of FTCs that may be represented by the MF-tire model are feasible, it cannot be assumed that every tire within the feasibility space can be manufactured. However, knowing in which regions of the design space no tire can be built adds a great deal of value to the top-down development process. In addition, based on existing measured tire data, the feasibility of the solution intervals for tires may be re-evaluated. Nevertheless, using existing tires only in order to identify the feasible space would lead to a too conservative feasible space, as it would exclude any progress in tire development. The feasibility prediction may be improved by use of more detailed subsystem models of the tire that compute the FTCs based on the design variables of the tire, e.g., chafer, rubber, bias, etc. Usually, more complex models result in an increased computational effort such that it would be vital for the applicability to find the right trade-off. Perhaps machine learning techniques can help to reduce the computational effort without neglecting the accuracy, as was previously introduced in this work for the suspension system.

Target Cascading Using Solution Spaces

For the first time, the methods introduced in this work took advantage of the benefits of solution spaces and target cascading. Therefore, the solution box on the subsystem level provides flexibility and robustness for the subsystem design, while the feasibility of the requirements is ensured.

Change in Mindset

Point-based optimization techniques propose only one particular design to the engineer and therefore are inferior compared to solution spaces. In contrast, the classical method for seeking solution spaces evaluates the biggest solution box as the best one under the given constraints. Therefore, solution spaces provide the engineer with multiple advantages regarding robustness and feasibility. However, if solution spaces are derived on complex subsystems, as it is the case in architecture design for vehicle dynamics, the largest solution box often is not feasible on a detail level due to certain restrictions. Therefore, the method introduced in this work does not simply seek the largest solution box but the largest solution box which is feasible on a detail level with a certain robustness. Results have proven that solution boxes that were optimized in terms of their size were not feasible and were then turned into feasible solution boxes based on the feasibility information provided from the detail level. As a result, detailed proposals for subsystem designs may be forwarded to the series development of the components, which was proven for, e.g., the suspension system.

Solution Spaces for Platform Design

Eichstetter (2017) and Erschen (2017) assume that components between different systems may be shared within the intersection of solution spaces. Therefore, the applied solution box is a subset of the intersecting solution space. A drawback of this method is that the remaining quantity of the solution spaces is lost as potential solution space for the commonal components. In addition, applying identical solution boxes is only one particular requirement among many others (e.g., geometric dimensions of an anti-roll bar) in order to establish commonality. Therefore, overlapping solution spaces are not sufficient in order to put the same component into a different vehicle. Considering more complex components or even subsystems such as the suspension system, current solution space approaches would lead to the necessity that all control links, the subframe as well as the wheel dimensions, are the same. Therefore, a great deal of potential solution space would not be used. The underlying reason is that solution spaces are derived on functional subsystem properties, since deriving them to the position of kinematic joint positions would result in very small solution spaces. However, those functional subsystem properties are influenced by multiple components that are intended to be shared between different vehicles. Therefore, commonality needs to be established on the detail level.

The approach introduced in this work extends the scope of considered boundary conditions (vehicle architecture, geometrical restrictions, causal relationships of complex subsystems) while focusing on establishing commonality of the components on the detail level. In addition, the architecture approach provides the possibility to balance the requirements between the different subsystems in order to release conflicting goals while making use of all the advantages of solution spaces. Therefore, components may be shared between different vehicles with fewer drawbacks in performance.

Components with Multiple Design Variables

If a particular component that is intended to be shared between different vehicles is described by more than one design variable, the associated design variables need to have the same value in the systems. Therefore, if solution spaces are applied, the solution interval of the design variables associated with the component need to be the same.

The *tire* is such a complex component that is treated on the subsystem level in this work. In order to use the same tire for different vehicles, not only do all FTCs need to be the same but also the dimensions of the tire, e.g., width and diameter.

To use the same *steering system* in separate vehicles, at least all associated design variables treated in this work on the subsystem level need to be the same. Nevertheless, there are several other properties of a steering system that are not considered in this work but should be in order to use the same, e.g., stiffness, screw attachment points etc.

The *front axle subframe* is another component described by multiple design variables. For the scope treated in this work, it is necessary that the relative distances between the associated kinematic joint positions in x- and z-direction are the same, and that the width is the same as well (y-position of the associated kinematic joint positions).

Subsystems with Multiple Components

If a subsystem consists of multiple components, establishing commonality on the detail/component level provides several advantages, e.g., different characteristics of the subsystem performance, which supports a different positioning of the vehicles within the markets. The *suspension system*, for example, is such a complex subsystem. Completely different subsystem performances may be achieved by a different arrangement of the same control links. Depending on the present architecture design, the required subsystem performance differs such that there is no better or worse performance in general. Instead, a suspension is good, if it suits the needs of the vehicle architecture.

Computational effort The major part of the overall computational effort results from the stochastic solution box optimization. Separate solution boxes for subsystems of different vehicles provide more opportunities in finding commonal solutions on the detail level. Consider two vehicles, one with a more direct steering feel than the other. If all other components except the suspension are the same, the steering feel needs to be adjusted by the suspension system. This may be achieved with a different yaw velocity response that is based on a different toe gradient over wheel travel that may be adjusted with the same components control links in with different alignment However, it increases the computational effort proportionally, as two solution box optimizations need to be performed for two vehicles instead of one optimization.

Objectives of the Work

In this subsection, whether the aims claimed in Section 1.3 were achieved is proven.

Aim 1: Providing a top-down method for the architecture design in vehicle dynamics that provides robustness and flexibility for the integration of the subsystems.

Based on existing literature, solution spaces provide multiple advantages compared to other top-down methods in terms of robustness and flexibility, e.g., no probability distributions are necessary, explicit threshold values for good and bad designs and also good visualization, see also Subsection 1.2.2. The application of this method to entire vehicle architectures was achieved by

- investigating the causal relationships and defining the relevant design variables (Section 2.1),
- enabling the tire model for architecture design (Chapter 3),
- creating fast computing neural networks that are able to reproduce the complex quantitative relationships of a vehicle architecture in a sufficient manner (Section 2.2),
- proposing an alternative stochastic optimization algorithm for solution boxes (Section 2.3).

The method was successfully applied in Chapter 6. Nevertheless, the proposed method is limited for two reasons.

One limiting factor is the accuracy of the applied neural networks, which gets worse in cases of increasingly complex[8] relations. Therefore, research regarding more advanced machine learning techniques or alternative approaches in order to reduce the computation time of a particular sample point would extend the applicability of the method more.

Another unsolved problem is the large number of computations for optimizing the solution box. However, so far only stochastic algorithms for deriving solution boxes are able to handle such complex problems.

Aim 2: Considering multiple feasibility restrictions of the particular subsystems in defining the requirements on the subsystems.

Based on a literature review, feasibility restrictions to solution spaces were not covered until now. In this work, different proposals for considering feasibility restrictions while deriving solution spaces were proposed for the first time.

Therefore, in order to establish the feasibility of the requirements derived to the *suspension system*, a robust optimization identifies a feasible design satisfying the requirements placed on the suspension system in a first step. Further on, if the requirements are satisfied, the subsystem is optimized for robustness in a second step. In the following iteration of the solution box optimization, the requirements previously placed on the subsystems are rebalanced in order to improve feasibility.

An alternative method for considering feasibility restrictions is applied to *tire* design. Compared to the design of the suspension system, no detailed simulation model is available such that the method proposed for the suspension system can not be used. Instead, a SVM is utilized for filtering non-feasible designs within the design space. Therefore, the algorithm of the solution box optimization is prevented from growing into those regions. Another advantage is that the user is able to distinguish between feasible and non-feasible regions directly by shifting the solution box within the design space, see Section 3.4.

Since the SVM classifier decides based on tires that may be represented by the MF tire model, the accuracy is limited. Therefore, it may be possible that not all tires within the identified feasible space are feasible indeed. However, the feasibility may be double checked based on the data of existing tires. The region where currently no tire exists, but tires may be represented by the MF tire model, is a grey zone where it cannot be clearly said whether a tire is feasible or not.

Aim 3: Deriving vehicle platforms on a detail level to ensure that different vehicles may share the same component.

Compared to existing methods that compute solution spaces in order to use common components, the methods presented in this work extends the scope of considered boundary conditions. The proposed methods allow multiple ways of considering vehicle platforms and

[8]Complex in terms of non-linearity, number of design variables, considered design space.

their typical constraints on the detail level. By considering multiple boundary conditions, the ability to share components between different vehicles is increased. Therefore, e.g., a tire, steering system, control link or front/rear axle subframe may be considered to be shared between different vehicles. The consideration of the detail level made it possible to design common components, even if they belong to a complex subsystem, without deriving solution spaces to the components directly, as this is not reasonable.

Although the proposed methods consider more details than already existing methods in the context of solution spaces, there are still more boundary conditions and design variables which are not considered. The virtual development of a vehicle is always a trade-off between the available quantitative models, their accuracy and computational expense, etc. Since the scope of this work is the design of vehicle dynamics, objectives regarding other fields such as durability, acoustics, ride comfort, etc. are not considered. As a result, there is a potential risk that, due to those neglected properties, it may not be possible to use a component of a vehicle platform.

CHAPTER 8

CONCLUSION AND OUTLOOK

In this chapter, a short summary of the results of this work is given and conclusions are made. Based on the conclusions drawn, future areas of research are identified.

8.1 Conclusion

The design of vehicle architectures and platforms is linked to many complex challenges, e.g., a large number of design variables and objective criteria, non-linearity of the causal relations and a large number of interdependencies between different parties. In addition, the development of new products is always associated with uncertainties due to, e.g., manufacturing tolerances or changing environmental conditions over the time of the development process. Overcoming them in an effective manner requires the use of virtual design approaches. However, virtual design approaches lead to additional uncertainties, e.g., in the determination of the objectives that satisfy the customers needs and the accuracy of the quantitative models. Therefore, the results of an appropriate method need to provide the engineer with flexibility in choosing a robust solution in order to reduce iteration loops, cost, and save time. As a result of the rising conflicting goals (e.g., weight, cost, performance, quality) and vehicle variety, the interplay between the particular subsystems and components need to be further improved to remain competitive. This in turn increases the considered complexity of the design problem.

In this thesis, a new consistent top-down development method for the design of vehicle architectures with respect to vehicle dynamics was introduced. To deal with the associated challenges, the solution spaces method presented by Zimmermann and Hoessle (2013) was extended by the consideration of feasibility restrictions. Therefore, the solution space method was integrated into a hierarchical target cascading approach.

Solution Spaces for Vehicle Architecture

Using solution spaces instead of particular design points provides the engineer with multiple advantages. By applying solution spaces, no specific robustness goal is necessary, as the engineer immediately sees the distance to the boundary of the solution space or solution box. As a result, the engineer is aware of the region within the design space where all top-level requirements placed on the vehicle are satisfied. The objective criteria for the overall vehicle that were considered in this thesis belong to the assessment fields of self-steering behavior, limit behavior, steering feel, parking, directional stability, steering power as well as others. The criteria were integrated into a holistic design method in order to consider them in the early development phase and to enable a fluent transition into the series production phase. Therefore, requirements were placed on objective criteria as upper and lower bounds. The actual vehicle performance of a particular design was measured during the simulation of open-loop maneuvers to evaluate if the overall vehicle targets were satisfied or not and to derive requirements on the essential subsystems. As a result, the driver was excluded from the driver/vehicle control loop. Nevertheless, depending on the target regions of the objective criteria, the customer experience is influenced, e.g., the steering feel may be tuned for specifically targeted customers. However, since the evaluation of steering feel is highly subjective and also influenced by the driver, the fine tuning will be done by experienced test drivers in the future as well.

If more than three design variables are considered, the complete solution space can no longer be visualized, therefore solution boxes are applied. Solution boxes offer another advantage as they decouple the design variables from each other. Since all possible parameter combinations of design variables within a solution box lead to a design that satisfies all overall vehicle objectives, they are treated as equal in terms of their performance. Therefore, the derived intervals of the solution box may be propagated to multiple departments in order to develop the subsystems in a decentralized manner with minimal coordination effort and friction.

Feasibility Restrictions

To use the potential of this decentralized development approach, the solution intervals forwarded to the different departments need to be trustworthy and therefore feasible by each department on its own, otherwise multiple meetings would be necessary in order to balance the requirements that were derived to the subsystems. The method introduced in this work considers feasibility restrictions while identifying a solution box. Therefore, an expandable hierarchical three-level approach was used considering the vehicle-DNA, tires, steering system, and suspension system as subsystems that need to be designed. As each subsystem has its own feasibility restrictions, three different methods for evaluating the feasibility were used. The applied methods are based on a robust optimization, SVM classifier and data-based approach of existing components.

For the *suspension system*, the feasibility was considered by a robust optimization between the subsystem and the detail level with the target to identify a design with its

performance within the required solution intervals and, if so, as close as possible to the center of the solution intervals. In a consecutive step, the solution box on the subsystem level is not only shifted regarding the overall vehicle targets but also the feasible design on the detail level. This procedure progresses until a solution box with a sufficient feasibility is found or a predefined maximum number of iterations is achieved.

For the *tire*, the feasibility was considered by overlaying the quantitative model that computes the system performance with a classifier that evaluates the feasibility of all sample points. Therefore, an SVM was trained based on tires that may be represented by the MF tire model which were created by DoE. In addition, the feasibility was crosschecked with measurements of existing tires from a database.

The presented methods for identifying the solution box with the biggest box size measure that is feasible is applicable to any non-linear, high dimensional design problem outside of the field of vehicle dynamics. However, the methods that are necessary for evaluating the feasibility may differ as we have seen in the examples of the suspension system and the tire.

Vehicle Platform Design

The new method introduced for the vehicle architecture design was also enabled to design components which are intended to be shared by different vehicles. Therefore, components that belong to a vehicle platform may be developed with respect to their needs. Contrary to other design approaches for vehicle platforms that use solution spaces, the method proposed in this work does not necessarily require overlaid solution boxes in order to be able to use the same component. Therefore, the advantages of separate solution boxes for each vehicle are still present, while they are optimized in order to use the same component one level beneath on the detail level. This is necessary, as it is not reasonable to derive solution boxes directly to the design variables of the subsystem, especially for complex subsystems such as the suspension system. Those subsystems are generally characterized by quite different configurations that meet the requirements, however the solution space in high dimensions looks more similar to a high dimensional curved tube than a box. For other subsystems, such as the tire, that seem to be able to be used in different vehicles as long as the solution box lies in the overlapping solution space of those vehicles, some further detail parameters, and boundary conditions, e.g., tire dimensions need to be considered as well. Since the tire dimensions impact the performance measures of the suspension system for a given set of kinematic joint positions, e.g., scrub radius and kinematic trail, they have a direct influence on the feasibility of the required suspension performance.

Application to Real World Design Problems

The methods presented in this work were applied to industrial design problems in the field of vehicle architecture design. Therefore, in the first example, multiple target ranges were defined for the overall vehicle with respect to vehicle dynamics in order to satisfy customer expectations. On the side of the design variables, the vehicle-DNA is already defined as

well as the rear suspension system. The tires are intended to be a carry-over part from an existing vehicle, which makes it difficult to satisfy the overall vehicle targets. Therefore, the relevant design variables are those of the front suspension system and the steering system. In a first iteration, the solution box with the biggest size measure was identified. Nevertheless, no feasible configuration of the kinematic joint positions that satisfies the performance requirements on the suspension system could be identified. Therefore, in the following iterations, the requirements on the front suspension system and the steering system were rebalanced with respect to the overall vehicle targets as well as the feasible front suspension performance. After six iterations, a consistent design providing a feasible solution box as well as a proposal for a robust design was identified. As a result, it may be obvious how important considering feasibility restrictions in vehicle architecture design really is.

Design Algorithms

The presented design algorithms provide vital support for the design of vehicles and vehicle platforms. However, their results depend on the following conditions:

- How well the entered targets reflect the desired vehicle behavior. If there is a breach between subjective evaluation and the objective evaluation the behavior of the vehicle will not satisfy the customer. The same applies if not all essential requirements were considered by the algorithm. However, this work sets the state-of-the-art in terms of considering an entire set of overall vehicle requirements in the field of vehicle dynamics.
- The best design algorithm is only as good as the underlying model. Earlier in this work, it was stated that the three tools on the way to architecture design build on each other. Therefore, if the model does not represent the reality in a sufficient manner, the results also will not. However, the presented algorithms can be applied independently of the underlying simulation models. The applicability rises and falls with the calculation time of the quantitative models, which depends on their complexity and the available computing power.
- Constraints concerning the vehicle platform, feasibility, etc. have been set realistically in order to avoid illusory solutions. The introduced methods focus on avoiding illusory solutions by evaluating the feasibility and enabling the user to implement various kinds of constraints. However, since the geometrical constraints for the kinematic optimization of the suspension are implemented by the user, an incorrect input can result in an infeasible design from a geometrical point of view, for example.

8.2 Outlook

Although a great amount of research and progress has been made in the field of systems design over the past years, it is still insufficient. Especially for complex systems (e.g.,

vehicles), each design method will reach its limits sooner or later. Therefore, only a part of the overall picture may be considered, which was the part of vehicle dynamics in this work. In the following, future areas of research are shortly summarized, based on the results of this thesis in order to increase the value added by system design approaches.

There is an *increased need to consider more complex simulation models* as a result of the rising demand for binding virtual development processes. A great challenge is to make them applicable for complex system design methods. However, they usually require more computational power due to the need for more computations. Therefore, any progress that leads to a relaxation of the trade-off between the accuracy of the provided results and computational effort may lead to a reduction of necessary hardware tests and development time. Initial approaches using machine learning techniques were presented in this thesis and may be extended by more advanced methods that require a deeper view on the latest innovations.

By using *two-dimensional-spaces* instead of solution boxes, a larger area of the solution space may be propagated to the different departments, which provides them with more flexibility and robustness. In the context of vehicle architecture design, a vital challenge will be to keep the advantage that the different departments are able to develop their subsystems largely independently of each other. In addition, the approach presented in Erschen 2017 needs to be extended in order to handle such complex design problems, e.g., by a stochastic algorithm.

The *prediction of the feasibility of the tire* may be even further improved. Therefore, simulation models that predict the FTCs of a tire based on physical and geometrical detail parameters (e.g., sidewall construction, rubber, etc.) may be necessary. In addition, the model needs to be designed in order to be combined with the design method for vehicle architectures. Based on the complexity of the model, the feasibility may be evaluated either by the method proposed for the suspension design or for the tire design.

If no feasible solution exists, whether for a particular vehicle or multiple vehicles that are intended to share components, requirements or boundary conditions need to be adapted. It may be time-consuming to adapt the overall vehicle goals with respect to the AIs and iterate towards a feasible solution. Therefore, some type of automated vehicle target generator/customizer would be helpful. This generator could check if the overall objectives on the vehicles are plausible and feedback the corresponding AIs.

Since a *hierarchical pure top-down development* method needs a certain number of iterations in order to identify a consistent design, the associated computational expense might be larger compared with other design methods (all-at-ones approaches). A valid approach to reduce the necessary effort to identify a consistent design would be to adapt the design variables of the x-level in order to identify a robust design on the y-level with respect to the overall targets on the z-level during the robust optimization.

$$\underset{x_1,x_2,\dots x_d}{\text{minimize}}\ \varphi_{\text{rb}}(\Omega_{\text{rb}}(\Omega^{\text{z}}_{\text{sb}})) \tag{8.1}$$

As a result, only designs that are feasible on the detail level are considered as possible solutions on the y-level. Therefore, feasibility may be established from the beginning on.

However, for the solution box optimization, this approach is not applicable, as a homogeneous sample inside the solution box is necessary. In addition, detailed quantitative models need to be available, which is not always the case in the early development phase. This in turn limits the applicability. However, the emphasis of this work was the introduction of a top-down method that may be extended as the timeline of a vehicle project progresses.

Integrating the *control systems* into the architecture design would offer two main advantages. First, considering them from the beginning of vehicle development would offer possibilities to reduce the costs of mechanical parts, if they can be compensated. Also, iteration loops for the application of the control systems may be reduced, as these are already pre-applied to the particular vehicles. Second, since the control systems may be adjusted accurately, they can be used in order to increase robustness or even to shift solution spaces to establish the use of commonal parts. Therefore, the introduced design algorithms would need to be adapted, e.g., by using them as additional design variables during the robust optimization without considering them in their condensed performance function.

APPENDIX

A.1 Accuracy of Quantitative Surrogate Models

This Section regards the evaluation of the accuracy of the quantitative surrogate models.

A.1.1 Vehicle Model

The accuracy of the artificial vehicle model may be quantified by the coefficient of determination R^2. An R^2 value of 1 means that the artificial model delivers the same output as the physical model for the tested input data, however, a value of 0 means that no correlation exists between both models. According to the R^2 value, both objective criteria have a good accuracy. Since a large amount of data was used to train and validate the model, the particular sample points in Figure A.1 are fused to a red region, although only the validation data is plotted. Therefore, the accuracy of $a_{\mathrm{y,\,max}}$ is better than one would estimate based on Figure A.1 (b).

As solution spaces only distinguish between good and bad designs, the RSM may also be evaluated by its ability to accurately separate the design space into regions with good or bad designs. Therefore a parameter set of a particular objective criterion is divided into the following four categories:

- True positive: designs with a performance value that is evaluated as good by the RSM and the original model.
- True negative: designs with a performance value that is evaluated as bad by the RSM and the original model.
- False positive: designs with a performance value that is evaluated as good by the RSM but bad by the original model. As a result, the solution box would be larger by use of the RSM.
- False negative: designs with a performance value that is evaluated as bad by the RSM but good by the original model. As a result, the solution box would be smaller by use of the RSM.

In practice, when the solution box is only allowed to contain good designs, its size is usually smaller if an RSM is used. Since only good designs are allowed to enter the solution box, a single design that is evaluated as *false negative* prevents the expansion of the solution box.

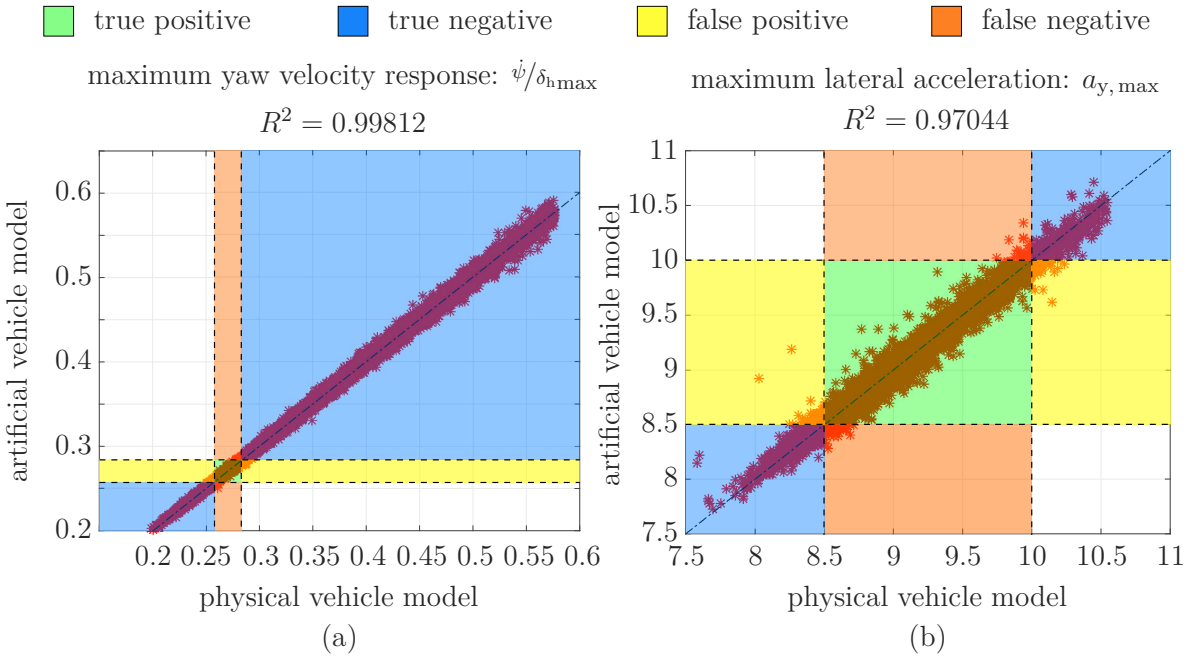


Figure A.1: Correlation between the output of the artificial vehicle model and the physical two-track model on the example of $a_{\mathrm{y},\max}$ and $\dot{\psi}/\delta_{\mathrm{h}\max}$.

The permissible intervals of the objective criteria in Figure A.1 and Figure A.2 are based on the application example in Section 6.5. Figure A.2 (a) shows that the percentage of correctly classified data (true positive, true negative) is much larger than the percentage of misclassified data (false positive, false negative). Therefore, as by the R^2 value, a good accuracy is indicated by evaluating the percentage of correctly classified data. However, the relation between the size of the permissible interval on the system level and the size of the design interval also has an impact on the required accuracy of the RSM. Figure A.1 (a) presents a relatively small permissible interval, while Figure A.1 (b) presents a big permissible interval. For an accurate position of the solution box, the relation between sample points that are evaluated as *true positive* and those that are misclassified (false positive, false negative) is important, see Figure A.2 (b). The smaller the percentage of design points evaluated as true positive, the more important it is to reevaluate the final design with the original model, especially if the design is close to such a particular threshold value. In general, Figure A.2 (a) indicates the accuracy of the solution box in relation to the design space while Figure A.2 (b) indicates the accuracy of the solution box in relation to the size of the solution box. Based on the R^2 value and the classical evaluation of binary classifiers in Figure A.2 (a), the RSM that computes $\dot{\psi}/\delta_{\mathrm{h}\max}$ is more accurate than $a_{\mathrm{y},\max}$. Nevertheless, in relation to the size of the associated permissible interval on the system level, $a_{\mathrm{y},\max}$ is more accurate than $\dot{\psi}/\delta_{\mathrm{h}\max}$.

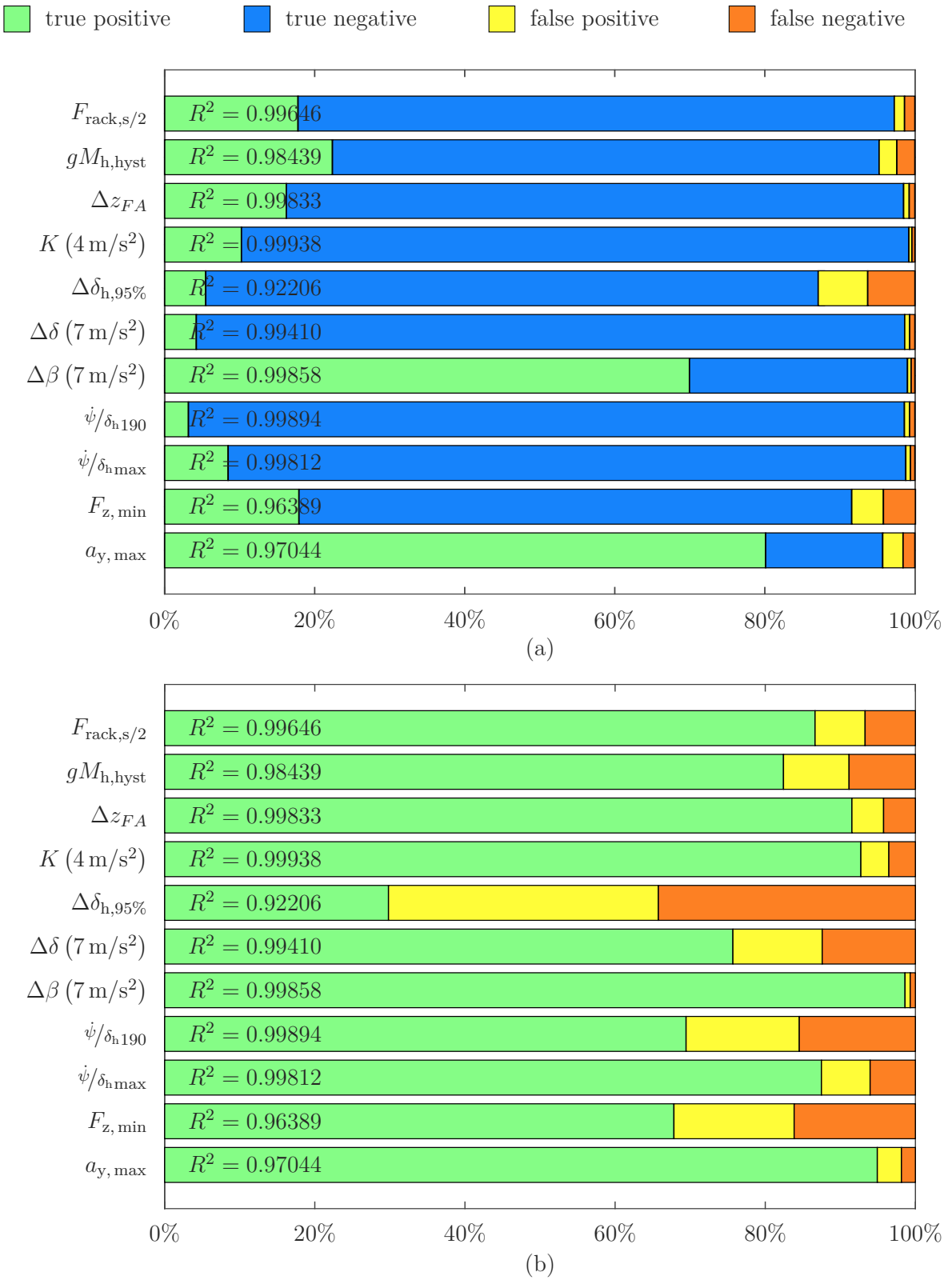


Figure A.2: Evaluation of the RSM quality for chosen objective criteria. (a) Misclassified data in relation to correctly classified data. (b) Misclassified data in relation to correctly classified data within a permissible interval.

A.1.2 Suspension Model

Using two different objective criteria as an example, Figure A.3 compares the accuracy of the artificial suspension model in relation to the physical suspension model. The R^2 is close to 1 for both objective criteria, which is good. The numbers next to the data points in Figure A.3 indicate the sample points with the largest deviation to the physical suspension model. Therefore, the engineer may prove if the associated input data for the simulation is valid. If not, the data can be filtered to improve the accuracy of the artificial model within the valid domain. In case of a high center of gravity, the vehicle might overturn before reaching the grip limit of the tire. Hence, the regression model is not able to represent such behavior.

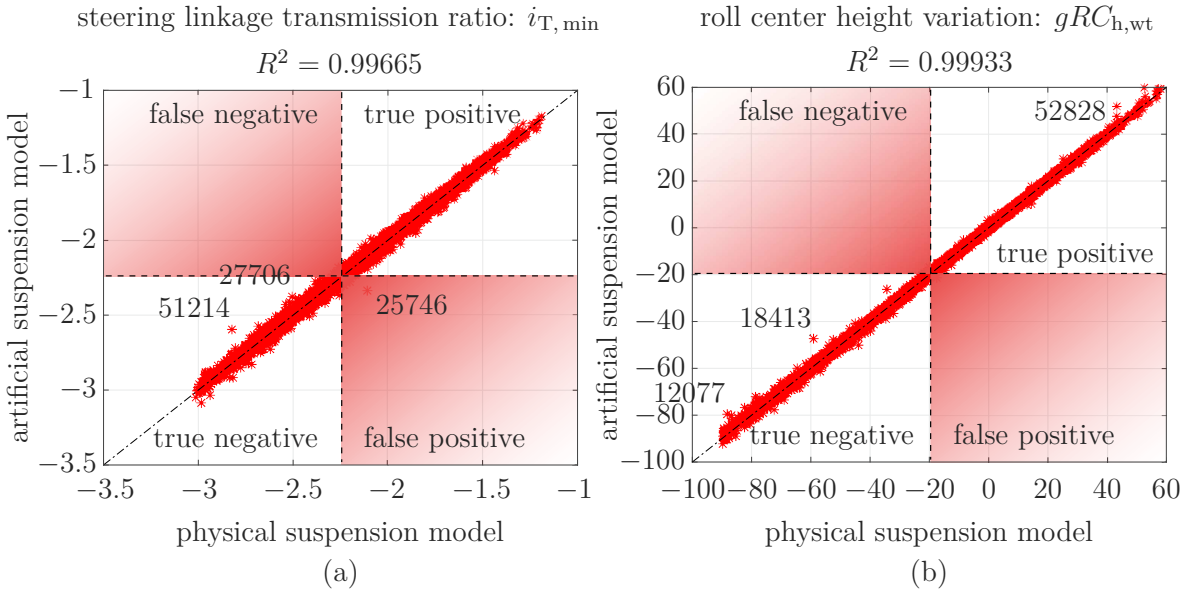


Figure A.3: Correlation between the output of the artificial suspension model and the kinematic suspension model on the example of $i_{\mathrm{T,\,min}}$ and $gRC_{\mathrm{h,wt}}$.

A.2 Geometrical Integration of a Suspension System

The example presented in Figure A.4 shows the importance of considering geometrical constraints during the feasibility optimization of a suspension system.

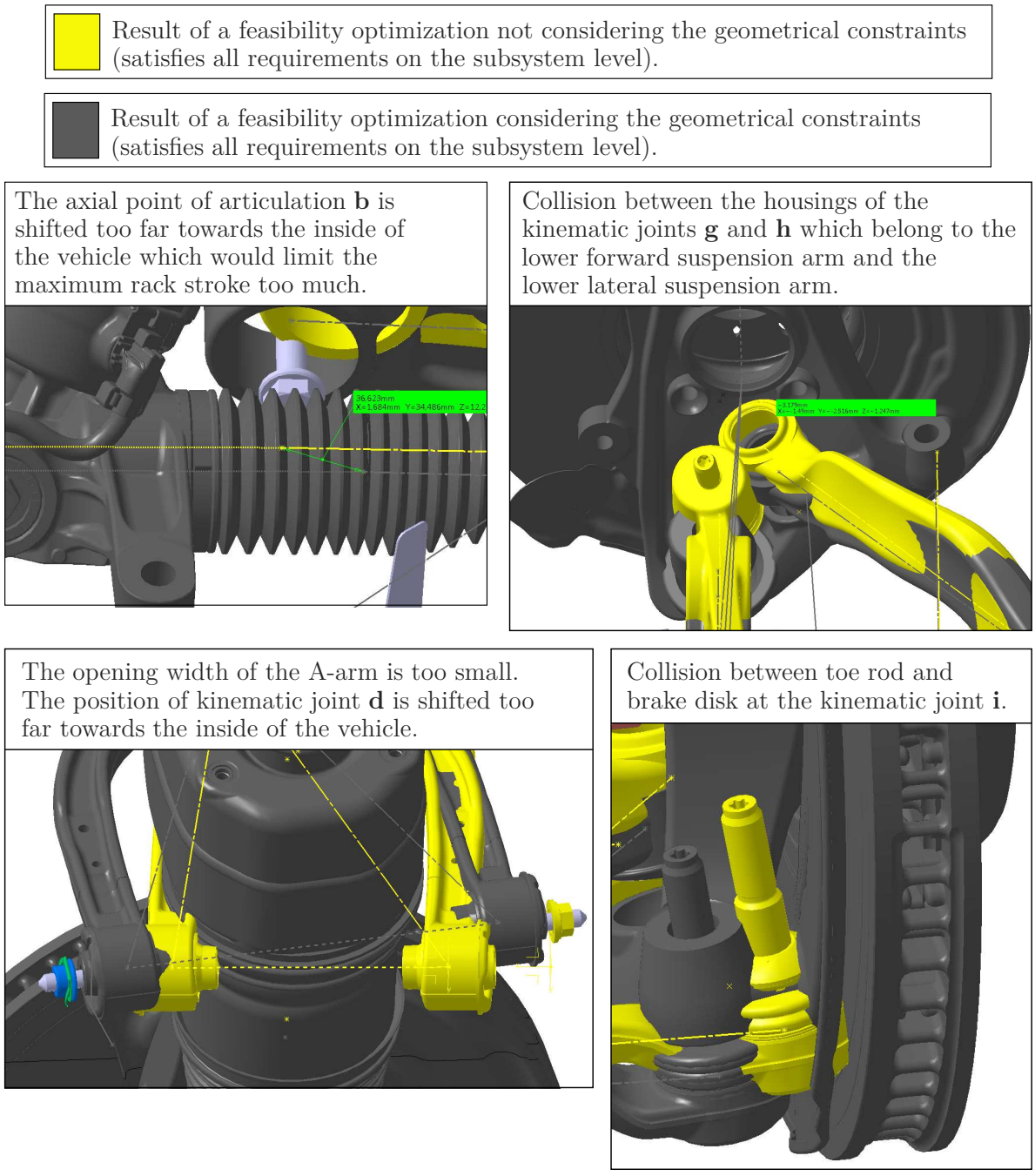


Figure A.4: Results of a feasibility optimization with (yellow) and without (grey) consideration of geometrical constraints.

A.3 Filter for Objective Quantities of the Vehicle

If the objective criteria of a Monte Carlo sampling are computed in order to generate a surrogate model, the measured performance needs to be filtered in order to obtain a sufficient accuracy. The percentage of sample points that are filtered depends on the size of the design space, the system response and the filter criteria. To create the artificial vehicle model, multiple filters were applied. For the objective criteria presented in Figure 6.7, the applied filter criteria are presented in Table A.1.

Table A.1: Filter criteria of objective criteria on the system level.

Filter $a_{\mathrm{y,\,max}}$ if:	
$a_{\mathrm{y,\,max}} < 7.5\,\mathrm{m/s^2}$	The maximum lateral acceleration is below a certain level.
$\delta_{\mathrm{h}}(a_{\mathrm{y,\,max}}) < \delta_{\mathrm{h}}(\mathrm{end})$	The steering angle at maximum lateral acceleration is smaller than at the end of the QSSC maneuver.
$M_{\mathrm{h}}(\mathrm{end}-1) < M_{\mathrm{h}}(\mathrm{end})$	The steering torque increases over the last simulation steps of the QSSC maneuver.
Filter $F_{\mathrm{z,\,min}}$ if:	
$F_{\mathrm{z,\,min}} < 25\,\mathrm{N}$	The minimum vertical tire load at maximum lateral acceleration is below a certain level.
$F_{\mathrm{z,\,min}} > 2500\,\mathrm{N}$	The minimum vertical tire load at maximum lateral acceleration is above a certain level.
$\delta_{\mathrm{h}}(a_{\mathrm{y,\,max}}) < \delta_{\mathrm{h}}(\mathrm{end})$	The steering angle at maximum lateral acceleration is smaller than at the end of the QSSC maneuver.
$M_{\mathrm{h}}(\mathrm{end}-1) < M_{\mathrm{h}}(\mathrm{end})$	The steering torque increases over the last simulation steps of the QSSC maneuver.
Filter $\dot{\psi}/\delta_{\mathrm{h\,max}}$ if:	
$\dot{\psi}/\delta_{\mathrm{h\,max}} < 0.1\,\mathrm{s^{-1}}$	The maximum yaw velocity response is below a certain level.
$\dot{\psi}/\delta_{\mathrm{h\,max}} > 1.5\,\mathrm{s^{-1}}$	The maximum yaw velocity response is above a certain level.
$\mathrm{numel}(\mathrm{findpeaks}(\dot{\psi}/\delta_{\mathrm{h}})) > 1$	The yaw velocity response has multiple local maximum values at different vehicle speeds.

Continued on next page

Table A.1 – continued from previous page.

Filter $\dot{\psi}/_{\delta_\mathrm{h}190}$ if:	
$\dot{\psi}/_{\delta_\mathrm{h}190} < 0.1\,\mathrm{s}^{-1}$	The yaw velocity response at 190km/h is below a certain level.
$\dot{\psi}/_{\delta_\mathrm{h}190\mathrm{km/h}} > 1.5\,\mathrm{s}^{-1}$	The yaw velocity response at 190km/h is above a certain level.
numel(findpeaks($\dot{\psi}/_{\delta_\mathrm{h}}$)) > 1	The yaw velocity response has multiple local maximum values at different vehicle speeds.
Filter $\Delta\beta\,(7\,\mathrm{m/s^2})$ if:	
$\Delta\beta\,(7\,\mathrm{m/s^2}) < 4°$	The body slip angle is below a certain level.
$a_{\mathrm{y,\,max}} < 7.5\,\mathrm{m/s^2}$	The maximum lateral acceleration is below a certain level.
Filter $\Delta\delta\,(7\,\mathrm{m/s^2})$ if:	
$\Delta\delta_\mathrm{h}\,(7\,\mathrm{m/s^2}) > 50°$	The steering angle is above a certain level.
$a_{\mathrm{y,\,max}} < 7.5\,\mathrm{m/s^2}$	The maximum lateral acceleration is below a certain level.
$\Delta\delta_{\mathrm{h},95\%} < 0°$	The difference between the steering wheel angle at 95% of the maximum lateral acceleration and a defined lower lateral acceleration is below a certain level.
Filter $\Delta\delta_{\mathrm{h},95\%}$ if:	
$\Delta\delta_{\mathrm{h},95\%} < 0°$	The difference between the steering wheel angle at 95% of the maximum lateral acceleration and a defined lower lateral acceleration is below a certain level.
$\Delta\delta_{\mathrm{h},95\%} > 25°$	The difference between the steering wheel angle at 95% of the maximum lateral acceleration and a defined lower lateral acceleration is above a certain level.
$a_{\mathrm{y,\,max}} < 8\,\mathrm{m/s^2}$	The maximum lateral acceleration is below a certain level.
Filter $gM_{\mathrm{h,hyst}}$ if:	
$gM_{\mathrm{h,hyst}} < 0.2\mathrm{Nm}$	The gradient of the steering wheel torque over lateral acceleration is below a certain level at $a_\mathrm{y} = 1\,\mathrm{m/s^2}$ in the WEAVE maneuver.
$gM_{\mathrm{h,hyst}} > 3\mathrm{Nm}$	The gradient of the steering wheel torque over lateral acceleration is above a certain level at $a_\mathrm{y} = 1\,\mathrm{m/s^2}$ in the WEAVE maneuver.
No filter is applied to: $K\,(4\,\mathrm{m/s^2})$, Δz_{FA} and $F_{\mathrm{rack,s/2}}$.	

BIBLIOGRAPHY

Al-Ashaab, A. et al. (2009). "Set-Based Concurrent Engineering Model for Automotive Electronic/Software Systems Development". In: *Proceedings of the 19th CIRP Design Conference*. Cranfield University.

Ao, S.-I., Rieger, B., and Amouzegar, M. A. (2010). *Machine Learning and Systems Engineering*. Springer Science+Business Media.

Automotive, T. (2008). *MF-Tool 6.1 Users Manual*.

Bai, F., Guo, K., and Lu, D. (2013). "Tire Model for Turn Slip Properties". In: *International Journal of Commercial Vehicles* 6.2, pp. 353–361.

Beyer, H.-G. and Sendhoff, B. (2007). "Robust Optimization - A Comprehensive Survey". In: *Computer Methods in Applied Mechanics and Engineering* 196 (33), pp. 3190–3218.

Bhattacharya, S., Krishnan, V., and Mahajan, V. (1998). "Managing New Product Definition in Highly Dynamic Environments". In: *Management Sience* 44 (11), pp. 50–64.

Binder, K. (2006). "Monte-Carlo Methods". In: *Mathematical Tools for Physicists*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 249–280.

Blundell, M. and Harty, D. (2004). *The Multibody Systems Approach to Vehicle Dynamics*. Elsevier Butterworth-Heinemann.

Boehm, B. W. (1984). "Verifying and Validating Software Requirements and Design Specifications". In: *IEEE Softw.* 1.1, pp. 75–88.

Bosch (2017). *Servoelectric Electromechanical steering system for a dynamic driving experience and highly automated functions*. URL: `http://www.bosch-automotive-steering.com/fileadmin/downloads/Flyer_Nkw/AS_Systemmappe_Servolectric_E_lowres_20150513.pdf`.

Box, G. and Fung, C. A. (1994). "Quality Quandaries* Is your robust design procedure robust?" In: *Quality Engineering* 6.3, pp. 503–514.

Braess, H. (2001). "Lenkung and Lenkverhalten von Personenkraftwagen -Was haben die letzten 50 Jahre gebracht, was kann and muss noch getan werden?" In: *VDI Berichte*.

Broy, M. and Rausch, A. (2005). "Das neue V-Modell XT". In: *Informatik-Spektrum* 28.3, pp. 220–229.

Bröhl, A.-P. and Dröschel, W. (1993). *Das V-Model - Der Standard für die Softwareentwicklung mit Praxisleitfaden*. R. Oldenbourg Verlag München.

Burges, C. J. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2.2, pp. 121–167.

Casalino, L. and Pastrone, D. (2015). "A Straightforward Approach for Robust Design of Hybrid Rocket Engine Upper Stage". In: *51st AIAA/SAE/ASEE Joint Propulsion Conference*.

Chandrasiri, N. P., Nawa, K., and Ishii, A. (2016). "Driving skill classification in curve driving scenes using machine learning". In: *Journal of Modern Transportation* 24.3, pp. 196–206.

Chen, X. et al. (1997). "Reliability based structural design optimization for practical applications". In: Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics.

Cheng, G., Xu, L., and Jiang, L. (2006). "A sequential approximate programming strategy for reliability-based structural optimization". In: *Computers & Structures* 84.21, pp. 1353–1367.

Cook, L. W., Jarrett, J. P., and Willcox, K. E. (2017). "Horsetail Matching for Optimization Under Probabilistic, Interval and Mixed Uncertainties". In: AIAA SciTech Forum. American Institute of Aeronautics and Astronautics.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Daub, M. and Duddeck, F. (2018). "Complex systems design under non-reducible lack-of-knowledge uncertainties". In: *ICVRAM ISUMA UNCERTAINTIES conference*.

Decker, M. (2009). "Zur Beurteilung der Querdynamik von Personenkraftwagen". PhD thesis. Germany: Technical University of Munich.

Doerry, N. (2012). "Modeling and Simulation Tools for Set-Based Design". In: NAFEMS.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley-Interscience.

Dudenhöffer, F. (2016). *Wer kriegt die Kurve? Zeitenwende in der Autoindustrie*. Campus Verlag.

Eichstetter, M. (2017). "Design of Vehicle System Dynamics using Solution Spaces". PhD thesis. Germany: Technical University of Berlin.

Eichstetter, M., Müller, S., and Zimmermann, M. (2015). "Product Family Design With Solution Spaces". In: *Journal of Mechanical Design* 137.12.

Eichstetter, M. et al. (2013). "Variance-based Sensitivity Analysis of a non-linear Two-Track Model for Vehicle Design in the early Conceptual Layout". In: *chassis.tech plus, 4th International Munich Chassis Symposium 2013*. Springer Fachmedien Wiesbaden.

Eichstetter, M. et al. (2014). "Solution Spaces for Damper Design in Vehicle Dynamics". In: *chassis.tech plus, 5th International Munich Chassis Symposium 2014*. Springer Fachmedien Wiesbaden, pp. 107–132.

Erschen, S. (2017). "Optimal Decomposition of High-Dimensional Solution Spaces for Chassis". PhD thesis. Technical University of Munich.

Erschen, S., Duddeck, F., and Zimmermann, M. (2015). "Robust Design using Classical Optimization". In: *PAMM - Procedings of Applied Mathematics and Mechanics* 15, pp. 565–566.

Ertel, W. (2013). *Grundkurs Künstliche Intelligenz. Eine praxisorientierte Einführung*. 3rd ed. Springer Fachmedien Wiesbaden.

Fellini, R. et al. (2001). "Target Cascading for Design of Product Families". In: *Proceedings of the 4th congress on structural and multidisciplinary optimization, Dalian, China*.

Fender, J. (2014). "Solution Spaces for Vehicle Crash Design". PhD thesis. Germany: Technical University Munich.

Fender, J. et al. (2014). "Identifying key parameters for design improvement in high-dimensional systems with uncertainty". In: *Journal of Mechanical Design* 4.

Finch, W. W. and Ward, A. C. (1997). "A Set-Based System for Eliminating Infeasible Designs in Engineering Problems dominated by Uncertainity". In: *Proceedings of DETC'97*. ASME Design Engineering Technical Conferences.

Forsberg, K. and Mooz, H. (1991). "The Relationship of System Engineering to the Project Cycle". In: *INCOSE International Symposium* 1.1, pp. 57–65.

Fritzsche, M. (2016). "Ein Beitrag zur Objektivierung des Sicherheitsempfindens mit Fahrermodellen bei Lenkstörungen". PhD thesis. Germany: University of Duisburg-Essen.

Ge, P., Lu, S. C.-Y., and Suh, N. P. (2002). "An Axiomatic Approach for "Target Cascading" of Parametric Design of Engineering Systems". In: *CIRP Annals - Manufacturing Technology* 51.1, pp. 111–114.

Gillespie, T. D. (1992). *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, Inc.

Gipser, M. (1999). "FTire, a New Fast Tire Model for Ride Comfort Simulations". In: *International ADAMS Users Conference*. Berlin.

Gong, W. et al. (2017). "Simplified-robust geotechnical design of soldier pile-anchor tieback shoring system for deep excavation". In: *Marine Georesources & Geotechnology* 35.2, pp. 157–169.

Graham, C. and Talay, D. (2013). "Strong Law of Large Numbers and Monte Carlo Methods". In: *Stochastic Simulation and Monte Carlo Methods: Mathematical Foundations of Stochastic Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 13–35.

Grosan, C. and Abraham, A. (2011). *Intelligent Systems, A Modern Approach*. Springer-Verlag Berlin Heidelberg.

Guiggiani, M. (2018). *The Science of Vehicle Dynamics. Handling, Braking, and Ride of Road and Race Cars*. 2nd ed. Springer International Publishing.

Guo, K. and Lu, D. (2007). "UniTire: Unified Tire Model for Vehicle Dynamic Simulation". In: *Vehicle System Dynamics* 45.sup1, pp. 79–99.

Guo, K., Lu, D., and Ren, L. (2001). "A unified non-steady non-linear tyre model under complex wheel motion inputs including extreme operating conditions". In: *JSAE Review* 22.4, pp. 395–402.

Hab, G. and Wagner, R. (2012). *Projektmanagement in der Automobilindustrie: Effizientes Management von Fahrzeugprojekten entlang der Wertschöpfungskette*. Springer Fachmedien Wiesbaden.

Haken, K.-L. (2015). *Grundlagen der Kraftfahrzeugtechnik*. Carl Hanser Verlag GmbH & Co. KG.

Harrer, M. and Pfeffer, P. (2016). *Steering Handbook*. Springer International Publishing.

Harrer, M. (2007). "Characterisation of Steering Feel". PhD thesis. United Kingdom: University of Bath.

Haskins, C., ed. (2006). *INCOSE Systems Engineering Handbook v. 3*.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*. Springer series in statistics. Springer.

Heissing, B., Ersoy, M., and Gies, S. (2011). *Chassis Handbook: Fundamentals, Driving Dynamics, Components, Mechatronics, Perspectives*. ATZ/MTZ-Fachbuch. Vieweg+Teubner Verlag, Wiesbaden.

Hirschberg, W., Rill, G., and Weinfurter, H. (2007). "Tire model TMeasy". In: *Vehicle System Dynamics* 45.sup1, pp. 101–119.

ISO 13674-1 (2003). *Road vehicles - Test method for the quantification of on-centre handling - Part 1: Weave test*. International Organization for Standardization.

ISO 19365 (2016). *Passenger cars - Validation of vehicle dynamic simulation - Sine with dwell stability control testing*. International Organization for Standardization.

ISO 26262 (2011). *Road vehicles – Functional safety*.

ISO 4138 (2012). *Passenger Cars - Steady-state circular dirving behaviour - Open-loop test procedure*. International Organization for Standardization.

ISO 7401 (2003). *Road vehicles - Lateral transient response test methods - Open-Loop test methods*. International Organization for Standardization.

Johri, K. (2019). *The Agile Concurrent Software Process*. URL: `https://www.softprayog.in/software-engineering/agile-concurrent-software-process`.

Kang, D. O., Heo, S. J., and Kim, M. S. (2010). "Robust design optimization of the McPherson suspension system with consideration of a bush compliance uncertainty". In: *Journal of Automobile Engineering*, pp. 705–716.

Kang, D. O. et al. (2011). "Robust design optimization of suspension system by using target cascading method". In: *International Journal of Automotive Technology* 13.1.

Kang, N. et al. (2012). "Optimal Design of Commercial Vehicle Systems Using Analytical Target Cascading". In: *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.

Kim, H. M. et al. (2003). "Analytical Target Cascading in Automotive Vehicle Design". In: *International Journal of Vehicle Design* 125, pp. 199–225.

Kim, H. M. (2001). "Target cascading in optimal system design". PhD thesis. USA: University of Michigan.

Kitayama, S. and Yamazaki, K. (2014). "Sequential approximate robust design optimization using radial basis function network". In: *International Journal of Mechanics and Materials in Design* 10.3, pp. 313–328.

Kokkolaras, M et al. (2002). "Extension of the target cascading formulation to the design of product families". In: *Structural and Multidisciplinary Optimization* 24.4, pp. 293–301.

Kokkolaras, M., Mourelatos, Z. P., and Papalambros, P. Y. (2004). "Design Optimization of Hierarchically Decomposed Multilevel Systems Under Uncertainty". In: *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 1, pp. 613–624.

Kracht, F. E. et al. (2015). "Development of a chassis model including elastic behavior for real-time applications". In: *chassis.tech plus, 6th International Munich Chassis Symposium 2015: chassis.tech plus*. Springer Fachmedien Wiesbaden, pp. 257–281.

Krenker, A., Bester, J., and Kos, A. (2011). *Introduction to the Artificial Neural Networks*. INTECH Open Access Publisher.

Krüger, H.-P. and Neukum, A. (2001). "Bewertung von Handlingeigenschaften — zur methodischen und inhaltlichen Kritik des korrelativen Forschungsansatzes". In: *Kraftfahrzeugführung*. Ed. by T. Jürgensohn and K.-P. Timpe. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 245–262.

Kuiper, E. and Van Oosten, J. J. M. (2007). "The PAC2002 advanced handling tire model". In: *Vehicle System Dynamics* 45.sup1, pp. 153–167.

Kvasnicka, P. et al. (2006). "Durchgangige Simulationsumgebung zur Entwicklung und Absicherung von Fahrdynamischen Regelsystemen". In: *Numerical analysis and simulation in vehicle engineering*, pp. 387–404.

Kvasnicka, P. and Schmidt, H. (2010). "Conceptual layout for spring/damper set up of a prototype regarding vehicle dynamics and ride comfort". In: *chassis.tech plus, 1st International Munich Chassis Symposium*, pp. 215–234.

Lehar, M. and Zimmermann, M. (2012). "An inexpensive estimate of failure probability for high-dimensional systems with uncertainty". In: *Structural Safety* 36 - 37, pp. 32 –38.

Liu, H. et al. (2005). "Probabilistic analytical target cascading - A moment matching formulation for multilevel optimization under uncertainty". In: *Journal of Mechanical Design* 128.

Lopez, R. H. and Beck, A. T. (2012). "Reliability-based design optimization strategies based on FORM: a review". en. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 34, pp. 506 –514.

Lunkeit, D. (2014). "Ein Beitrag zur Optimierung des Rückmelde- und Rückstellverhaltens elektromechanischer Servolenkungen". PhD thesis. Germany: University of Duisburg-Essen.

Ma, B. et al. (2016). "Analysis of vehicle static steering torque based on tire-road contact patch sliding model and variable transmission ratio". In: *Advances in Mechanical Engineering* 8.9.

Mareis, T. (2014). *TRW stellt neue Version seiner elektrischen Servolenkung vor*. URL: `https://www.krafthand.de/artikel/trw-stellt-neue-version-seiner-elektrischen-servolenkung-vor-14976/`.

Mastinu, G., Gobbi, M., and Miano, C. (2006). *Optimal Design of Complex Mechanical Systems. With Applications to Vehicle Engineering*. Springer Verlag, Berlin Heidelberg.

McCulloch, W. S. and Pitts, W. (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Michelena, N., Kim, H. M., and Papalambros, P. (1999). "A System Partitioning And Optimization Approach To Target Cascading". In: *INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN*.

Michelena, N., Park, H., and Papalambros, P. Y. (2003). "Convergence properties of analytical target cascading". In: *AIAA Journal* 41.5, pp. 896–905.

Michelena, N. et al. (2001). "Design of an Advanced Heavy Tactical Truck: A Target Cascading Case Study". In: *SAE Technical Paper Series* 2001-01-2793.

Michie, D., Spiegelhalter, D. J., and Taylor, C. (1994). *Machine Learning, Neural and Statistical Classification*. E. Horwood.

Morokoff, W. J. and Caflisch, R. E. (1994). "Quasi-Random Sequences and Their Discrepancies". In: *SIAM Journal on Scientific Computing* 15.6, pp. 1251–1279.

Mäder, D. (2012). "Simulationsbasierte Grundauslegung der Fahrzeug-Querdynamik unter Berücksichtigung von Erfahrungswissen in der Fahrdynamikentwicklung". PhD thesis. Germany: Technical University of Kaiserslautern.

Müller, K.-R. et al. (2001). "An introduction to kernel-based learning algorithms". In: *IEEE Transactions on Neural Networks* 12.2, pp. 181–201.

Münster, M. et al. (2014). "Vehicle steering design using solution spaces for decoupled dynamical subsystems". In: *International Conference on Noise and Vibration Engineering, ISMA*.

Na, J. and Gil, G. (2016). "Virtual Optimization of Tire Cornering Characteristics to Satisfy Handling Performance of a Vehicle". In: *SAE Technical Paper Series* 2016-01-1652.

Niedermeier, F., Peckelsen, U., and Gauterin, F. (2013). "Virtual tires in the early vehicle development phase". In: *International Journal of Emerging Technology and Advanced Engineering* 3.3, pp. 394–401.

Niedermeier, F. et al. (2013). "Virtual Optimization of the Interaction between Tires and the Vehicle". In: *International Journal of Modeling and Optimization* 3.1, pp. 20–24.

Pacejka, H. B. (2006). *Tyre and Vehicle Dynamics*. Automotive engineering. Butterworth-Heinemann.

Pacejka, H. B. and Bakker, E. (1992). "The Magic Formula Tire Model". In: *Vehicle System Dynamics* 21, pp. 1–18.

Panchal, J. H. et al. (2005). "An Interval-Based Focalization Method for Decision-Making in Decentralized, Multi-Functional Design". In: *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 31st Design Automation Conference. Vol. 2, pp. 413–426.

Pande, P., Neuman, R., and Cavanagh, R. (2000). *The Six Sigma Way: How GE, Motorola, and Other Top Companies are Honing Their Performance*. McGraw-Hill Education.

Park, G.-J. et al. (2006). "Robust Design: An Overview". In: *AIAA Journal* 44.1, pp. 181–191.

Park, H.-U. et al. (2015). "Uncertainty-based MDO for aircraft conceptual design". In: *Aircraft Engineering and Aerospace Technology* 87.4, pp. 345–356.

Rocca, E. and Timpone, F. (2016). "Analysis of Multilink Suspension Characteristics due to the Rods Length Variations". In: *Proceedings of the World Congress on Engineering 2016 Vol II*.

Rousseeuw, P. (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20, pp. 53–65.

Roy, B. K. and Chakraborty, S. (2015). "Robust optimum design of base isolation system in seismic vibration control of structures under random system parameters". In: *Structural Safety* 55.Supplement C, pp. 49 –59.

Röski, K. (2012). "Eine Methode zur simulationsbasierten Grundauslegung von PkW-Fahrwerken mit Vertiefung der Betrachtungen zum Fahrkomfort". PhD thesis. Germany: Technical University of Munich.

Sancibrian, R. et al. (2010). "Kinematic design of double-wishbone suspension systems using a multiobjective optimisation approach". In: *Vehicle System Dynamics* 48.7, pp. 793–813.

Schramm, D. et al. (2017). *Fahrzeugtechnik: Technische Grundlagen aktueller und zukünftiger Kraftfahrzeuge*. De Gruyter Studium. Oldenbourg.

Schramm, D., Hiller, M., and Bardini, R. (2018). *Vehicle Dynamics. Modeling and Simulation*. Springer Berlin Heidelberg.

Shahraki, A. F. and Noorossana, R. (2014). "Reliability-based robust design optimization: A general methodology using genetic algorithm". In: *Computers & Industrial Engineering* 74.Supplement C, pp. 199 –207.

Sharon, T. (2018). *Continuous user research in 11.6 seconds*. URL: `https://medium.com/@tsharon/continuous-user-research-in-11-6-seconds-57c27d1595c5`.

Sobek II, D. K., Ward, A. C., and Liker, J. K. (1999). "Toyota's Principles of Set-Based Concurrent Engineering". In: *Sloan Management Review* 2 (40), pp. 67–83.

Song, L., Fender, J., and Duddeck, F. (2015). "A Semi-analytical Approach to Identify Solution Spaces for Crashworthiness in Vehicle Architectures". In: *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV): Traffic Safety Through Integrated Technologies*.

Stapelberg, R. F. (2009). *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*. Springer London.

Suh, N. P. (2001). *Axiomatic Design: Advances and Applications*. Oxford University Press.

Suh, N. P. (1998). "Axiomatic Design Theory for Systems". In: *Research in Engineering Design* 10.4, pp. 189–209.

Suh, N. P.e. a. (1979). "Application of Axiomatic Design Techniques to Manufacturing". In: *The American Society of Mechanical Engineers for presentation at the Winter Annual Meeting*.

Taguchi, G., Chowdhury, S., and Wu, Y. (2004). *Taguchi's Quality Engineering Handbook*. 1st ed. Wiley-Interscience.

The Law of Large Numbers and the Monte Carlo Method (2014). University of Massachusetts Amherst. URL: `https://people.math.umass.edu/~lr7q/ps_files/teaching/math456/lecture17.pdf`.

The MathWorks Inc. (2016). *Introducing Machine Learning*.

Thomke, S. H. (1997). "The role of flexibility in the development of new products, An empirical study". In: *Research Policy* 26.1, pp. 105 –119.

TNO (1999). *TIME - Tire Measurement Procedure*.

Uselmann, A. (2017). "Ein Beitrag zur funktionalen Entwicklung eines elektromechanischen Lenksystems für sportlich orientierte Fahrzeuge". PhD thesis. Germany: University of Duisburg-Essen.

V-Modell-Autoren (2017). *V-Model XT (Version: 2.1) - Das deutsche Referenzmodell für Systementwicklung*.

Ward, A. C. (1989). "A theory of quantitative inference applied to a mechanical design compiler". PhD thesis. USA: Massachusetts Institute Of Technology.

Wimmler, J. et al. (2015). "Concurrent Design of Vehicle Tires and Axles". In: *chassis.tech plus, 6th International Munich Chassis Symposium*. Springer Fachmedien Wiesbaden.

Wimmler, J. et al. (2016). "Optimizing suspension systems for feasibility, performance and cost". In: Crc Press, pp. 229–234.

Wolf, H. J. (2009). "Ergonomische Untersuchungen des Lenkgefühl an Personenkraftwagen". PhD thesis. Technical University of Munich.

Wolf-Monheim, F. (2014). "CAE-based driving comfort optimization for passenger cars". In: *chassis.tech plus, 5th International Munich Chassis Symposium 2014: chassis.tech plus*. Wiesbaden: Springer Fachmedien Wiesbaden, pp. 133–149.

Wu, S. et al. (2009). "Special Analytical Target Cascading for Handling Performance and Ride Quality Based on Conceptual Suspension Model and Multi-body Model". In: *SAE Technical Paper Series* 2009-01-1455.

Yang, R. J. and Gu, L. (2004). "Experience with approximate reliability-based optimization methods". In: *Structural and Multidisciplinary Optimization* 26.1, pp. 152–159.

Yi, P. and Cheng, G. (2008). "Further study on efficiency of sequential approximate programming for probabilistic structural design optimization". In: *Structural and Multidisciplinary Optimization* 35.6, pp. 509–522.

Zhang, X. et al. (2008). "Steering feel study on the performance of EPS". In: *Vehicle Power and Propulsion Conference*. IEEE.

Zhao, Y.-G. and Ono, T. (1999). "A general procedure for first/second-order reliabilitymethod (FORM/SORM)". In: *Structural Safety* 21.2, pp. 95–112.

Zimmermann, M. and Hoessle, J. E. (2013). "Computing solution spaces for robust design". In: *International Journal for Numerical Methods in Engineering* 94.3, pp. 290–307.

Zimmermann, M. et al. (2017). "On the design of large systems subject to uncertainty". In: *Journal of Engineering Design* 28.4, pp. 233–254.