Application of Optimization Methods to Controller Design for Active Suspensions

Von der Fakultät für Maschinenbau, Elektrotechnik und Wirtschaftsingenieurwesen der Brandenburgischen Technischen Universität Cottbus zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation vorgelegt von

Tuan-Anh Nguyen, M.Sc.,

geboren am 03.01.1972 in Hanoi, Vietnam

Vorsitzender: Prof. Dr.-Ing. **Peter Steinberg**
Gutachter: Prof. Dr.-Ing. habil. **Dieter Bestle**
Gutachter: Prof. Dr.-Ing. **Gerhard Lappus**

Tag der mündlichen Prüfung: 06.10.2006

Preface

The work described in this thesis was carried out at the Brandenburg University of Technology, Cottbus, at the Chair of Engineering Mechanics and Vehicle Dynamics between April 2003 and October 2006.

First and foremost I would like to express my deepest gratitude to Professor Dieter Bestle, my research supervisor and reviewer, for whom I have the greatest amount of respect and admiration. Professor Dieter Bestle not only afforded me the opportunity to study at the BTU Cottbus, but also provided valuable education and direction in all fields related to this research. Equally, I am indebted to Professor Gerhard Lappus who did a thorough revision of the work as second reviewer.

It is a great pleasure to work with the colleagues at the Chair. My special thanks to Lutz Anklam for supporting the computer facility. I would like to thank Andrej Wachal for helping me to become familiar with the MATLAB optimization toolbox. I thank Marcin Beffinger for the valuable discussions on vehicle dynamics. I sincerely thank all the members of the Chair of Engineering Mechanics and Vehicle Dynamics for creating such a relaxed and supportive working environment.

This research was funded by the Ministry of Education and Training of Vietnam. I thank this organization, without whose financial support this work would not have been possible.

Finally, my thanks go to my parents and my wife for the continuous support they gave while the thesis was taking shape.

Tuan-Anh Nguyen, October 2006.

Contents

Preface ii

Notation and Acronyms vi

1 Introduction 1

1.1 Literature Survey ……. 4
1.1.1 Passive, semi-active and active suspensions …….. 4
1.1.2 Vehicle modeling …… 7
1.1.3 Control algorithms for active suspensions …….. 8
1.1.4 Multi-criterion optimization …… 9
1.1.5 Gain-scheduling control …….. 10
1.2 Outline of the Dissertation ……..... 11

2 Multibody System Dynamics 13

2.1 Multibody Systems ……..……..……..……..... 13
2.2 Kinematics of Multibody Systems ……..…….. 15
2.3 Kinetics of Multibody Systems (Newton-Euler Equations) ……..….. 16
2.4 Reduction and Linearization of the Equations of Motion ……...…….. 18

3 Passenger Car Modeling 21

3.1 Suspension Forces ……..…. 21
3.1.1 Double-control-arm suspension ...……..……..… 21
3.1.2 Modified suspension parameters …….. 29
3.2 Three Degree-of-freedom Spatial Car Model ……..……..… 31
3.3 Plane Track Model …….…….....……..……..……..… 33
3.4 Linearized Equations of Motion of the Spatial Car Model ……..…… 35
3.5 State-space Representation of the Spatial Car Model …… 38

3.5.1 General applied forces ……..……. 38
3.5.2 Linear parameter-varying spatial car model ……..…. 42
3.6 Simulation Model of the Spatial Car ……..…….…..… 45

4 Linear Quadratic Regulator (LQR) Control 46

4.1 LQR Problem for Linear Systems without Disturbances ……..…….. 46
4.1.1 Definition of the LQR problem ……..……..…..… 47
4.1.2 LQR solution using Pontryagin's maximum principle ……..…..… 48
4.1.3 Algebraic Riccati equation ……..……..……..…. 51
4.2 LQR for Linear Systems with Measurable Disturbances …… 53
4.2.1 Problem definition ……..……..…… 53
4.2.2 Solution based on Pontryagin's maximum principle ……..…..… 54
4.3 Application of LQR Control to the Spatial Car Model …….……...….. 58
4.3.1 Dynamic criteria for the spatial car model …….……... 58
4.3.2 Spatial car model simulation ……..……...… 59
4.3.3 LQR design for the spatial car model ……..…… 60

5 Multi-criterion Optimization (MCO) 64

5.1 Overview on Multi-criterion Optimization ……..…..……...…. 64
5.2 Multi-criterion Optimization Methods ……..……..……..…… 68
5.2.1 Compromise method ……..……..……..…….. 68
5.2.2 Recursive knee approach ……..……..……..…… 70
5.3 MCO Problem for Passive Suspension ……...……..……..…... 73
5.3.1 Problem definition ……..……...……..…….. 73
5.3.2 Optimization results based on compromise method …….…... 73
5.4 MCO Problem for Active Suspension ……..……..……..... 76
5.4.1 Problem definition ……..……..……..…….. 76
5.4.2 MCO with LQR control ……..……...…..……. 77
5.4.3 Optimization results based on compromise method ……...…. 79
5.4.4 Optimization results based on recursive knee approach …….…. 82
5.5 MCO Problem for Passive and Active Suspensions ……..……...….. 85
5.5.1 Problem definition ……..……...……..…….. 85
5.5.2 Optimization results …….…... 86

6 Gain-scheduling Control 89

6.1 Operating Region of the Car Model ……. 89
6.1.1 Vehicle sideway-slipping and rollover stability ……. 90
6.1.2 Effects of suspension and tire compliance ……..… 91
6.1.3 Definition of operation region and operation points ……..... 93
6.2 Gain-scheduling Control Design for the Spatial Car Model …...…… 95
6.2.1 Local optimal controllers ……..……..… 95
6.2.2 Parameter-dependent controller design ……..…….. ……..…. 98
6.3 Vehicle Handling Test Simulation ….…… 101
6.3.1 Double-lane-change maneuvers ...……..……..... 101
6.3.2 Path generation problem for double-lane-change maneuvers ……. 103
6.3.3 Simulation results with gain-scheduling control ……..…... 110

7 Summary 113

Bibliography 115

Appendices 123

A Parameters of the Spatial Car Model ……. 123
B NEWEUL Output File for the Spatial Car Model ……..... 125
C MATLAB S-Function for Simulation ……..….. 133
D MATLAB.m Files for Optimization ……..... 136
D.1 Suspension Optimization ……..…... 136
D.2 Path Generation for Double-Lane-Change Maneuver ……..……..... 139

Notation and Acronyms

Mathematical Symbols

$\forall$	for all
$\setminus$	not including, except
■	end of proof
$\exists$, $\nexists$	there exists, exists not
$\in, \notin$	element of, not element of
$\subset, \not\subset$	subset of, not subset of
$\wedge, \vee$	logical *and*, *or*
$\{a_1, a_2, \ldots, a_n\}$	set of elements $a_1, a_2, \ldots, a_n$
$\{a \mid \ldots\}$	set of all a satisfying conditions …
$\mathcal{A} \cap \mathcal{B}$	intersection set of $\mathcal{A}$ and $\mathcal{B}$: $\mathcal{A} \cap \mathcal{B} = \{a \mid a \in \mathcal{A} \wedge a \in \mathcal{B}\}$
$\mathcal{A} \cup \mathcal{B}$	union set of $\mathcal{A}$ and $\mathcal{B}$: $\mathcal{A} \cup \mathcal{B} = \{a \mid a \in \mathcal{A} \vee a \in \mathcal{B}\}$
$\mathcal{A} \setminus \mathcal{B}$	$\mathcal{A}$ without $\mathcal{B}$: $\mathcal{A} \setminus \mathcal{B} = \{a \mid a \in \mathcal{A},\ a \notin \mathcal{B}\}$
$\mathcal{I}_n$	set of n integers $\{1, 2, \ldots, n\}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of n-dimensional real vectors
$\mathbb{R}^{n \times m}$	set of $n \times m$ matrices with elements from $\mathbb{R}$
$\lvert a \rvert$	absolute value of a
$a := b$,	a defined as b
$a \stackrel{!}{=} b$	a has to be identical to b
$a \geq b$, $a \leq b$	a is larger or equal, smaller or equal to b
$a << b$	a is much smaller than b
$a \approx b$	a is approximately b
$i = j\ (k)\ n$	i runs from j to n with steps k
$diag\{a_1, a_2, \ldots, a_n\}$	$n \times n$ diagonal matrix with $a_1, a_2, \ldots, a_n$ on its diagonal
$\max\{a_1, a_2, \ldots, a_n\}$	maximum value among a set of elements $a_1, a_2, \ldots, a_n$
$\min\{a_1, a_2, \ldots, a_n\}$	minimum value among a set of elements $a_1, a_2, \ldots, a_n$

da	total differential of a
∂a	partial differential of a
δa	variation of a
$\dot{a}$	first derivative of a with respect to time, $\dot{a} = da/dt$
$\ddot{a}$	second derivative of a in time, $\ddot{a} = da^2/d^2t$
a'	derivative of a with respect to a variable depending on the context
$\boldsymbol{a}$, a_i	column vector in matrix and index notation
$\boldsymbol{a}^T$, a_i	row vector
$\tilde{\boldsymbol{a}}$	skew-symmetric matrix of vector $\boldsymbol{a}$
$\boldsymbol{A}$, A_{ij}	matrix
$\boldsymbol{A}^T$, A_{ij}^T	transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$, A_{ij}^{-1}	inverse matrix, $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$
$\boldsymbol{0}$	zero vector, zero matrix
$\boldsymbol{A} > 0$	positive definite matrix, i.e. $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0 \ \forall \ \boldsymbol{x} \neq \boldsymbol{0}$
$\boldsymbol{A} \geq 0$	positive semi-definite matrix, i.e. $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq 0 \ \forall \ \boldsymbol{x}$
$\boldsymbol{A} = \boldsymbol{A}^T$	symmetric matrix
$\boldsymbol{I}$	identity matrix

Latin letters

a_x, a_y	$[m/s^2]$	longitudinal and lateral acceleration
$a_{y\,roll}$	$[m/s^2]$	lateral acceleration at rollover threshold
$a_{y\,slip}$	$[m/s^2]$	lateral acceleration at sideway-slipping threshold
a_y^u	$[m/s^2]$	upper bound of lateral acceleration
$\boldsymbol{a}$, $\bar{\boldsymbol{a}}$		translational and local translational acceleration vector
b	$[Ns/m]$	modified suspension damping coefficient
b_0	$[Ns/m]$	original suspension damping coefficient
b_f, b_r	$[Ns/m]$	damping coefficients of front and rear suspension
f		degree-of-freedom of position
f_i^*		individual minimum of criterion function f_i
f_S	$[N]$	total suspension force
fz_C	$[N]$	applied force in car body fixed frame
$\boldsymbol{f}$		vector of criteria

$\boldsymbol{f}^a, \boldsymbol{f}^r$		vector of applied and reaction forces
$\boldsymbol{f}_A, \boldsymbol{f}_B, \boldsymbol{f}_K$		vector of anti-roll, damper and spring forces
$\boldsymbol{f}_S$		vector of total suspension forces
g	[m/s^2]	gravity acceleration (9.81 m/s^2)
$\boldsymbol{g}$		vector of generalized constraint forces
$\boldsymbol{g}$		vector of equality constraints
h		number of design variables
h_C	[m]	static height of the car's center of gravity above ground
h_R	[m]	height of suspension roll center above ground
h_{RC}	[m]	distance from roll axis to the car's center of gravity
h_{RV}	[m]	distance from ground to roll axis
$\boldsymbol{h}$		vector of excitation forces
$\boldsymbol{h}$		vector of inequality constraints
k	[N/m]	translational stiffness
k_0	[N/m]	original suspension stiffness
k_f, k_r	[N/m]	stiffness of front and rear suspension
k_x		component of state-feedback gain matrix
k_w		component of disturbance-feed forward gain matrix
l		number of equality constraints
l_f, l_r	[m]	distance from the car's center of gravity to front and rear axles
l_β, l_α	[Nm]	applied roll and pitch moment in the car body fixed frame
$\boldsymbol{k}$		vector of Coriolis and centrifugal forces
$\boldsymbol{l}^a, \boldsymbol{l}^r$	[Nm]	vector of applied, reaction moments
m		number of constant curvatures of vehicle path model
m_i	[kg]	mass of body i
m_S	[kg]	sprung mass
m_U	[kg]	unsprung mass
n		number of criteria
n		number of points changing curvature of vehicle path model
$\boldsymbol{n}$		normal vector of convex hull of the individual minima
p		number of rigid bodies
$\boldsymbol{p}_0$		initial point design
$\boldsymbol{p}^*$		optimal design
$\boldsymbol{p}^{EP}$		Edgeworth-Pareto optimal design
$\boldsymbol{p}^l, \boldsymbol{p}^u$		vector of lower and upper variable bounds

q		number of independent constraints
$\boldsymbol{q}$		vector of generalized applied forces in inertial coordinate system
$\boldsymbol{q}^a$		vector of generalized applied forces in the car body fixed frame
$\bar{\boldsymbol{q}}^a$, $\bar{\boldsymbol{q}}^c$		global vector of applied forces, Coriolis and centrifugal forces
r	[*N/m/rad*]	modified rotational stiffness of anti-roll bar
r_0	[*Nm/rad*]	original anti-roll bar rotational stiffness
r_f, r_r	[*N/m/rad*]	rotational stiffness of front and rear anti-roll bar
r_k		rate of curvature change
$\boldsymbol{r}$	[*m*]	position vector
s	[*m*]	track coordinate
t	[*s*]	time
t_k	[*s*]	time point of changing curvature
t_w	[*m*]	mean value of half-track width
t_{wf}, t_{wr}	[*m*]	half track width of front and rear axle
u	[*N*]	control force
u_0	[*N*]	original control force
$\boldsymbol{u}$	[*N*]	vector of controlled forces
$\boldsymbol{u}_x$, $\boldsymbol{u}_w$	[*N*]	state-feedback and disturbance feed-forward control
v	[*m/s*]	longitudinal velocity
$\boldsymbol{v}$, $\bar{\boldsymbol{v}}$		translational and local translational velocity
w_i		weighting factor of criterion i
$\boldsymbol{w}$		vector of exogenous disturbances
x, y, z		axes of reference frame
x_{wi}, y_{wi}	[*m*]	longitudinal and lateral position of the i^{th} wheel in car body system
$\boldsymbol{x}$		state vector
$\boldsymbol{y}$		vector of generalized coordinates
$\boldsymbol{y}$		vector of measured outputs
z_C	[*m*]	vertical displacement of the car's center of gravity
z_S	[*m*]	vertical displacement of sprung mass
z_{Si}	[*m*]	vertical displacement of the i^{th} suspension node
z_U	[*m*]	vertical displacement of unsprung mass
$\boldsymbol{z}$		vector of generalized velocities
$\boldsymbol{z}_S$		vector of vertical displacement of suspension nodes
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{B}_w$		state matrices
$\boldsymbol{B}_S$		suspension damping matrix

$\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{D_w}$		measurement matrices
$\boldsymbol{D}$		matrix of damping forces
$\mathcal{F}$		feasible criterion space
$\mathcal{F}^{EP}$		Edgeworth-Pareto optimal set in design space
$F_{z\text{-in}}$, $F_{z\text{-out}}$	$[N]$	total normal load on inside and outside wheels
$F_{y\text{-in}}$, $F_{y\text{-out}}$	$[N]$	total lateral force on inside and outside wheels
F_A, F_B, F_K	$[N]$	anti-roll, damper and spring force
F_C	$[N]$	control force
F_U	$[N]$	anti-roll force acting on unsprung mass
$\boldsymbol{F}$		translational distribution matrix
$\boldsymbol{G}$		matrix of gyroscopic forces
$\mathcal{H}$		Hamiltonian function
I_X, I_Y, I_Z	$[kg\ m^2]$	roll, pitch and yaw moment of inertia of car body
$\boldsymbol{I_i}$		inertia tensor of body i
J		objective function
$\boldsymbol{J_T}$, $\boldsymbol{J_R}$		translational and rotational Jacobi matrix
$\overline{\boldsymbol{J}}$		global Jacobi matrix
$\boldsymbol{K}$		stiffness matrix
$\boldsymbol{K}$, $\boldsymbol{K_x}$		gain matrix of state-feedback controller
$\boldsymbol{K_w}$		gain matrix of disturbance-feed forward controller
$\boldsymbol{K_A}$		anti-roll stiffness matrix
$\boldsymbol{K_S}$		suspension stiffness matrix
$\boldsymbol{L}$		rotational distribution matrix
M_U	$[Nm]$	anti-roll moment
$\boldsymbol{M}$, $\overline{\overline{\boldsymbol{M}}}$		mass matrix, global mass matrix
$\boldsymbol{N}$		matrix of non-conservative forces
$\mathcal{P}$		feasible design space
$\mathcal{P}^{EP}$		Pareto frontier or trade-off curve of non-dominated solutions
$\boldsymbol{P}$		matrix of velocity-dependent forces
$\boldsymbol{P}$		Riccati matrix
$\boldsymbol{Q}$		matrix of position-dependent forces
$\boldsymbol{Q}$, $\boldsymbol{Q_y}$		weighting matrix of states and measured outputs
$\overline{\boldsymbol{Q}}$		global distribution matrix of reaction forces
R	$[m]$	track curve radius
$R_{\min}$	$[m]$	minimum turning radius

$\boldsymbol{R}$, $\boldsymbol{R_u}$		weighting matrix of controlled forces
$\boldsymbol{S}$		rotation matrix
T	[s]	simulation time
X_{wi}, Y_{wi}	[m]	longitudinal and lateral position of the i^{th} wheel in inertial system
X_V, Y_V	[m]	longitudinal and lateral position of car in inertial coordinate system
Y_w^l, Y_w^u	[m]	lower and upper bound on wheel tracks
$Y_{\text{cone}}^l, Y_{\text{cone}}^u$	[m]	lateral position of lower and upper cones

Greek letters

α	[rad]	car body pitch angle
$\boldsymbol{\alpha_i}$, $\overline{\boldsymbol{\alpha}}_i$		angular and local angular acceleration vector of body i
β	[rad]	car body roll angle
$\ddot{\beta}$	[rad/s^2]	car body roll acceleration
γ	[rad]	yaw angle
$\dot{\gamma}^u$	[rad/s]	uper bound on yaw rate
δ	[m]	displacement of suspension connecting points
δW	[J]	virtual work
ε_i		constraint on criterion f_i for compromise method
θ	[rad]	slope angle of car body roll axis
κ	[m^{-1}]	track curvature
κ_j	[m^{-1}]	constant path curvature
λ		modified coefficient of suspension parameters
$\boldsymbol{\lambda}$		dynamic Lagrange multipliers
ξ	[rad]	angular displacement of unsprung mass
ξ_0	[rad]	directional angle of lower control arm
$\boldsymbol{\xi}$		Riccati vector
φ_0, φ_{C0}	[rad]	directional angle of spring-damper and actuator
μ_y		road friction coefficient
$\boldsymbol{\omega_i}$, $\overline{\boldsymbol{\omega}}_i$		angular and local angular velocity vector of body i
Δl	[m]	dynamic deflection of spring-damper
Δz	[m]	relative vertical displacement of sprung and unsprung mass

List of Acronyms

ARE	Algebraic Riccati Equation
CG	Center of Gravity
CHIM	Convex Hull of Individual Minima
CUSC	Consumers Union Short Course
DRE	Differential Riccati Equation
DAE	Differential-Algebraic Equations
DLC	Double-Lane-Change
DoF	Degree-of-Freedom
EP	Edgeworth-Pareto
ISO	International Organization for Standardization
LQR	Linear Quadratic Regulator
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MBS	Multi-Body System
MCO	Multi-Criterion Optimization
NBI	Normal-Boundary Intersection
ODE	Ordinary Differential Equations
RMS/r.m.s	Root Mean Square

Chapter 1

Introduction

Demands for better ride comfort, road handling and controllability of passenger cars have motivated automotive industries to consider the use of active and semi-active suspensions. Many analytical and experimental studies on active and semi-active suspensions to improve ride quality and handling performance have been performed. The conclusion is that active and semi-active suspensions can provide substantial performance improvements over passive suspensions in general.

The effectiveness of the active suspension system on vehicle dynamics is analyzed based on vehicle models. Passenger cars are complex multibody systems consisting of many rigid and deformable components, Popp and Schiehlen [59], Rahnejat [60], Rill [63] and Willumeit [92]. A full vehicle model needs to present the nonlinear kinematics of wheels and axles, the effects of suspension geometry and has to include the drive train, the steering mechanism and the tire dynamics, Kortüm and Lugner [42], resulting in a high number of degrees of freedom. Since it makes no sense to try to build a universal vehicle model that can be used to solve all dynamic problems, reduced dynamic models for specific investigation purposes are often designed instead, Eberhard and Schiehlen [24] and Rettig and Stryk [62]. The vehicle yaw dynamics is mainly studied based on the conventional planar models such as single track model, Ammon [2], Mitschke [51] and Wallentowitz [87], or double track model, Ackermann [1], Halfmann and Holzmann [36] and Kiencke and Nielsen [40], where the effects of active suspensions are not taken into consideration. On the other hand, yaw motion is usually neglected when the quarter-car, half-car or spatial-car model with active or semi-active suspensions are investigated. In order to study the effects of active suspensions on the vehicle yaw dynamics, a proper mathematical model of the vehicle must be established that can describe the dynamic characteristics of interest sufficiently, but at the same time can be easily treated in control synthesis.

The application of active suspensions involves indispensably the application of control algorithms. Active control concepts have been investigated extensively over the past ten years. The purpose of an optimal control problem is to determine the control policy optimizing specific criteria, subject to the constraints imposed by the physical nature of the problem. One of the most effective optimal control techniques commonly used in engineering is the linear quadratic regulator (LQR) control, Colaneri, et al. [14], Ramirez [61], Schwarz [71] and Siouris [75]. For linear systems whose states can be measured online, the LQR algorithm results in a simple state-feedback control structure and provides an easy way to derive the optimal controller. By choosing appropriately the weighting factors corresponding to the criteria of interest, the optimal controller can be immediately obtained via MATLAB software. However, the standard LQR problem is defined only for linear systems without disturbances. In practice, dynamic systems usually are affected by exogenous excitations. Therefore, a control law with state-feedback and disturbance-feed forward controllers has been applied by several investigators as an optimal solution for the disturbance-rejection control problem, e.g. Ackermann [1], Bail [4] and Sampson [67]. Different formulas for the controllers have been applied without theoretical basis. Therefore, developing the LQR problem for excited linear-systems to affirm the control law as well as to define correctly the optimal controllers is necessary.

Another approach to achieve the optimal controller is multi-criterion optimization (MCO). Once the control structure is determined, the optimal components of the controller can be obtained from the associated multi-criterion optimization problem. Furthermore, the best trade-off between conflicting criteria can be also derived. In order to find the optimal compromise solutions, which are known as the Edgeworth-Pareto (EP) optimal solutions, the multi-criterion optimization problem usually has to be reduced to scalar utility problems, Bestle [7], Das [18] and Eberhard, et al. [22]. Being one of the effective scalarization approaches, the compromise method can generate an even distribution of the EP-optimal solutions on the trade-off surface even if the criterion space is non-convex, Collette and Siarry [16] and Deb [20]. This method, however, results in wasted computational resources to problems with more than two criteria. To deal with such problems, the recursive knee approach introduced by Das and Denis [19] and Wachal and Bestle [86], an advanced optimization method that can create a representative set of the EP-optimal solutions with a minimal computation effort, should be applied.

Solving the multi-criteria optimization problem directly for the optimal controller would be time-consuming, especially for large systems where the number of controller components often is cumbersome. This problem can be solved by the combination of the LQR algorithm, i.e., instead of finding directly the components of the controllers, the task of optimization routine is to define the optimal weighting factors of the associated LQR problem. For each time simulation, the controller results from the LQR algorithm based on the weighting factors provided by optimization algorithm. By combining multi-criterion optimization with the LQR control, not only computational time can be reduced significantly, but also the limitation of the LQR algorithm to the constrained control problems can be overcome.

A constant optimal controller obtained from the LQR-based optimization method proposed above, however, is valid only to a specific operation point defined by specific values of the system parameters. Vehicle dynamic systems usually include parameters that can be changed arbitrarily by different drive maneuvers or road conditions, thus resulting in parameter-varying systems. To maintain the desired performances of a parameter-varying system, the controller has to be able to change its parameters corresponding to the change of the system varying parameters over their operation regions, Sastry and Bodson [68]. The process for designing such controllers for parameter-varying systems is referred to as gain-scheduling control design. Recently, various gain-scheduling design techniques have been introduced. These methods, however, require either a complicated control structure, see e.g. Balas, et al. [5] and Genc [32], or a complex computation procedure, see e.g. Ackermann [1]. Therefore, a strategy for designing gain scheduling based on the defined optimal control structure and the proposed optimization method must be studied.

The effectiveness of the designed gain-scheduling controller must be evaluated through vehicle handling test maneuvers. It is well-known that there are an infinite number of paths that could satisfy the requirements for the double-lane-change maneuver at a given speed, which is designated as the standard vehicle handling test. In order to find an optimal path with respect to specific requirements, the path generation problem must be formulated. There are several different objectives for path optimization such as optimization of driving time, deviation from the lane center and driving safety, O'Hara [54]. Aiming to define an optimal path that enhances driving safety, the vehicle lateral dynamics can be reduced by minimizing track curvature, and thus minimizing lateral acceleration during the test.

1.1 Literature Survey

The description given above shows that, active suspension design relies on the fields of vehicle dynamic modeling, optimal control, multi-criterion optimization, and gain-scheduling control. The following sections briefly describe papers relevant and complementary to this research.

1.1.1 Passive, semi-active and active suspensions

The purpose of an automobile suspension is to adequately support the chassis, to maintain tire contact with the ground, and to manage the compromise between vehicle road handling and passenger comfort. Depending on the configurations and implementations, vehicle suspension systems can be classified as passive suspension, semi-active suspension or active suspension.

When designing a passive suspension, the trade-off mentioned above is made upfront and cannot be easily changed. For example, a sports car suspension will have stiffer shock absorbers for better road handling while the shock absorbers on a family vehicle will be softer for a comfortable ride. In the case of semi-active and active suspension systems, the trade-off decisions can be changed in real-time.

A semi-active suspension has the ability to change the damping characteristics of the shock absorbers (dampers) by continuously varying intensity of a magnetic field, Figure 1.1a, e.g. Choi, et al. [13], Genç [31], Paré [56] and Spencer, et al. [76], or by regulating a controllable orifice, Krüger [43]. As for a passive damper, the applicable force in a semi-active damper depends on the sign of the stroke velocity across the damper. Since semi-active dampers can only dissipate energy, not every control command can be applied. As shown in Figure 1.1b, only forces lying in the first and third quadrant of the force-stroke velocity plane can be produced, i.e. a positive force F_C can only be supplied while the damper is compressing and a negative force while expanding. If the controller commands a negative force during damper compression, the best that can be done is to generate only a compression force as small as possible, in other words, to set the current input to zero.

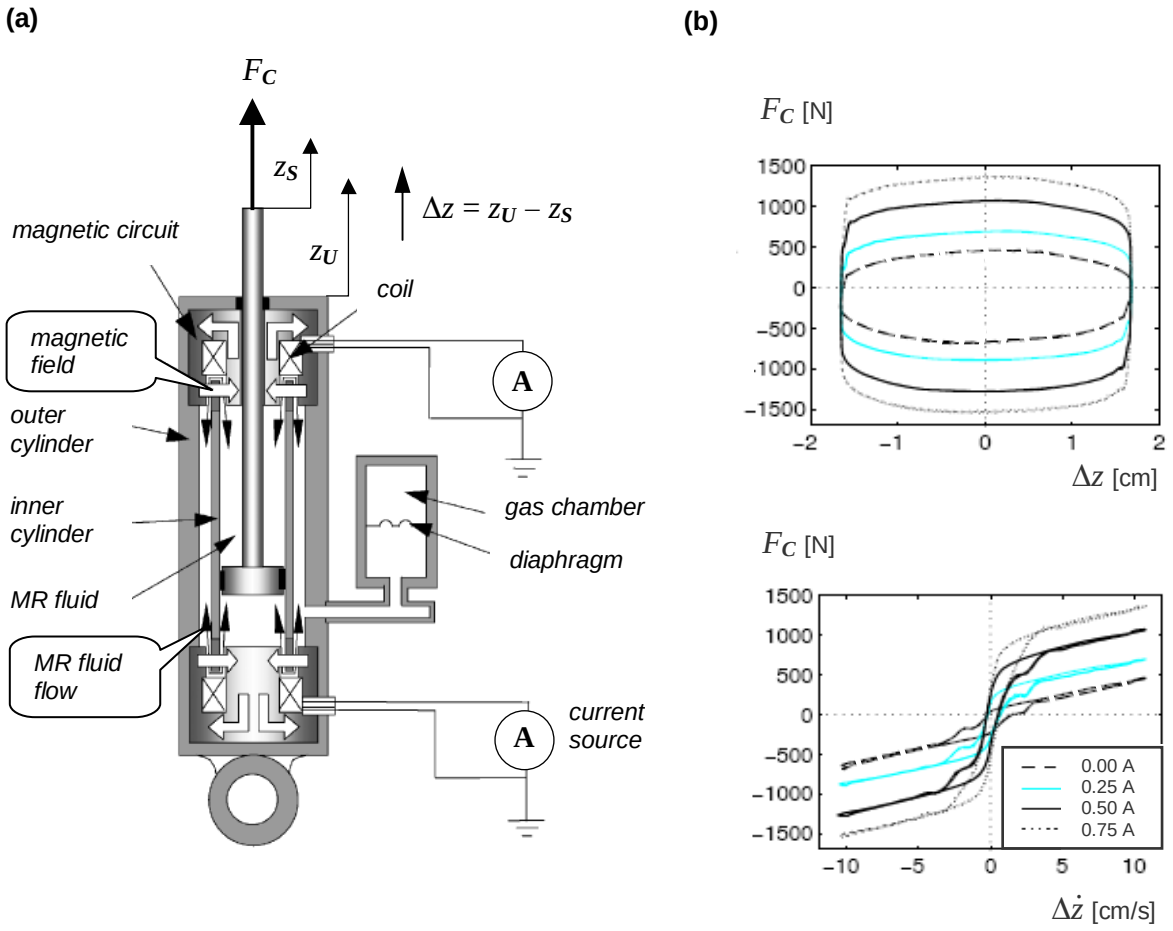

Figure 1.1: Schematic configuration **(a)** and characteristics **(b)** of magnetorheological (MR) dampers for different currents

Contrary to semi-active suspensions, hydraulic actuators of fully active suspensions can generate continuously controlled forces, i.e. they can both add and dissipate energy from the system, and thus provide better performance than semi-active suspensions. The hydraulic actuators are typically governed by electro hydraulic servo-valves and are mounted in parallel to passive suspension springs and dampers, allowing for the generation of forces between the sprung and unsprung masses. The electro hydraulic system consists of an actuator, a primary power spool valve and a secondary bypass valve. As seen in Figure 1.2, the hydraulic actuator cylinder lies in a follower configuration to a critically centered electro hydraulic power spool valve with matched and symmetric orifices. Positioning of the spool z_{sp} directs high pressure fluid flow to

either one of the cylinder chambers and connects the other chamber to the pump reservoir. This flow creates a pressure difference ΔP across the piston which acts on the piston to provide the active force F_C for the suspension system. The change in force is proportional to the position of the spool with respect to center which is controlled by a current-position feedback loop, the relative velocity of the piston, and the leakage through the piston seals, Donahue [21]. The research represented in this thesis assumes fully active suspensions.

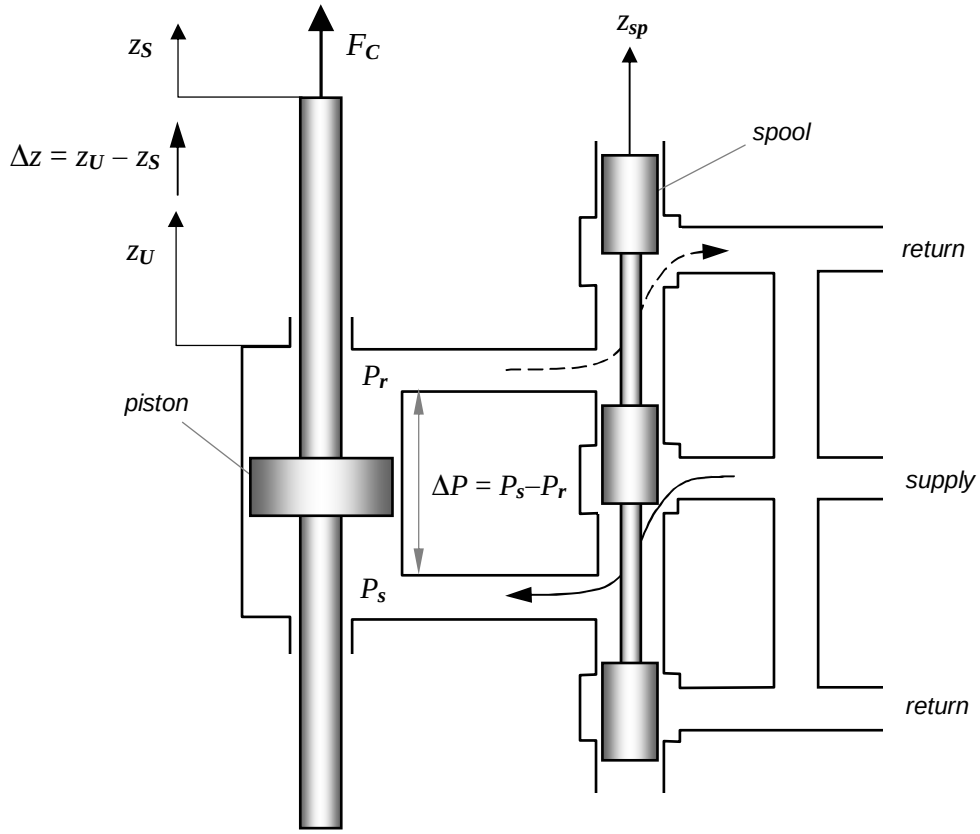

Figure 1.2: Schematic configuration of hydraulic actuator

A complete suspension typically consists of passive components and an actuator. Most technical solutions use the actuator in parallel to a conventional spring and damper, as illustrated in Figure 1.3 for a quarter-car model, for reasons of safety, i.e. to guarantee vehicle stability in case of actuator failure, and energy savings.

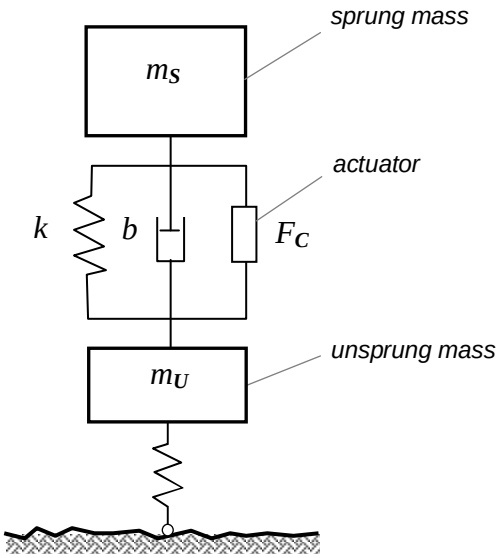

Figure 1. 3: Quarter-car model with active suspension

1.1.2 Vehicle modelling

Physical models for investigating the vertical dynamics of suspension systems are most commonly built on the conventional *quarter-car model*, which represents the vertical motion of a system including a quarter of the car body and the corresponding wheel, e.g. Chantranuwathana and Peng [10], Donahue [21], Pang, et al. [55], Shen and Peng [73] and Yi and Song [94]. To take into account the suspension geometry, Hong, et al. [38] introduced a *plane quarter-car model* with a semi-active Mac-Pherson suspension. More accurate analysis is achieved by extensions to a so-called *full-car model*, e.g. Choi, et al. [13] and Park and Kim [57], which reflects both vertical deflections and inclinations. Bounce, roll and pitch motions of the car body can be investigated simultaneously. In addition, the effects of suspension geometry and stabilizers or anti-roll bars also can be involved in the model, e.g. Gärtner and Saeger [30] and Mitschke [50, 51]. Separated and decoupled investigations are possible using *half-car models*, e.g. Gaspar, et al. [29], Taghirad and Esmailzadeh [78] and Vaughan [83].

The most commonly used models for studying vehicle lateral dynamics are the conventional planar models such as *single track model*, e.g. Ammon [2], Lazic [45], Lu, et al. [46]

and Ryu [65], and *double track model*, e.g. Ackermann [1], Halfmann and Holzmann [36] and Kiencke and Nielsen [40]. Although the yaw motion is taken into account, the suspension effects are not considered for these models. Hyvärinen [39], Sampson and Cebon [66] and Sampson [67] investigated the effects of the suspension system on vehicle lateral dynamics based on a *half-car roll model*. Additionally, the influences of the suspension and tire deformations on the vehicle stability and handling were also evaluated by Bodie and Hac [8] and Hac [34, 35].

1.1.3 Control algorithms for active suspensions

One of the most straightforward and effective control approaches for active suspensions is the so-called *sky-hook control*, which is used to hang up the vehicle body on a virtual sky completely uncoupled from road excitations. A large number of applications in the literature exist which often consist the skyhook approach as the reference control law; many of those investigations have used the quarter car model as a basis, e.g. Choi, et al. [13], Donahue [21] and Krüger [43]. Analogously, the *ground-hook control* concept takes into account wheel oscillations, e.g. Valasek, et al. [82].

Linear quadratic regulator (LQR) is a powerful concept of optimally controlling linear systems commonly used for vehicle system control. This technique results in a simple control structure with an optimal state-feedback controller which can easily be obtained from the solution of the algebraic Riccati equation. Several applications of the LQR control have been used in active suspension control, e.g. Rettig and Stryk [62], Sampson [67] and Taghirad and Esmailzadeh [78].

For complex systems where not all states are accessible to be measured, *Kalman filter techniques* are often used, Moscinski and Ogonowski [52] and Shahian and Hassul [72]. The LQR control with Kalman filter has been applied in the investigations of Krüger [43], Venhovens and Nabb [84] and Yi and Song [94]. Another approach proposed by Vaughan [83] is to use the *LQR control with output-feedback* controller.

Dealing with the uncertainties in system parameters, many robust control techniques have been developed. The most commonly used is the H_∞ *control*, e.g. Choi, et al. [13], Gaspar, et al. [29] and Wu [93]. Additionally, adaptive extensions to the standard LQR

control have been performed, Chantranuwathana and Peng [10]. Beside that, there also exists a variety of alternative formulations of the problem to control active suspension systems such as *fuzzy logic control*, e.g. Krüger [43] and Rouieh and Titli [64], and *sliding mode control*, e.g. Chen and Huang [11], Yokoyama, et al. [95] and Zhong [96].

1.1.4 Multi-criterion optimization

As already mentioned, suspension design has to resolve the conflict between ride safety and ride comfort resulting in a multi-criterion optimization problem. There exist a large number of methods and algorithms for solving such multi-criterion optimization problems, see for example Andersson [3], Coleman, et al. [15], Collette and Siarry [16], Deb [20] and Marler and Arora [47]. Most methods attempt to scalarize multiple objectives and perform repeated applications to find a set of Edgworth Pareto (EP)-optimal solutions, Bestle [7] and Shukla and Deb [74].

Aiming to provide a good diversity among solutions in the criterion space, beside the compromise method various advanced algorithms have been developed. The first one is the *normal boundary intersection* (NBI) method, developed by Das [17, 18] and Das and Dennis [19]. Their study was aimed at getting a good diversity of solutions on the efficient frontier by starting from normal directions to the ideal plane passing through individual function minimizers. The study used an equality constraint formulation of the sub-problems. A modified version of the NBI approach, called the *recursive knee approach*, was developed by Das and Dennis [19]. Better formulations were also introduced and programmed by Wachal and Bestle [86].

Kim and Weck [41] developed the *adaptive weighted-sum method* for multi-criterion optimization. Initially, the efficient frontier is approximated by employing a single-objective optimization algorithm with the weighted-sum approach many times. Efficient front patches are then identified and further refined by using additional equality constraints.

Mattson, et al. [48] and Messac, et al. [49] developed the *normal constraint method* for getting an even distribution of the EP-optimal solutions on the Pareto frontier. In the normal constraint method, there is a sequential reduction of the feasible space by hyper-planes passing through a point on the ideal plane. Chen, et al. [12] also developed the

physical programming method and then presented a different method for generating the entire efficient frontier using the physical programming approach.

Over the past decade, the *evolutionary multi-objective optimization* received growing attention by its ability for finding multiple EP-optimal solutions in a single simulation run and providing the entire range of solutions and the shape of the Pareto frontier, Deb [20] and Shukla and Deb [74]. Applications of the evolutionary multi-objective optimization to the design of rail vehicle suspensions performed by Eberhard, et al. [23] and He [37] demonstrated the effectiveness of this method.

1.1.4 Gain-scheduling control

Due to arbitrary changes of the system parameters resulting from different drive maneuvers or road conditions, vehicle dynamic systems are often formulated as parameter-varying systems which require the controller to change its parameters appropriately. Gain-scheduling is one of the most intuitive approaches to adaptive control, commonly used to control linear parameter-varying (LPV) systems. This technique amounts to design controllers which are able to update their parameters on-line according to the variations of the system parameters. The advantage of gain-scheduling is that the required performances of the system are guaranteed by the rapid change of the control parameters in response to the changes in the system dynamics, Sastry and Bodson [68].

Conventionally, gain-scheduling control is designed by a two-step procedure: first one designs local controllers at specified operation points, then a parameter-dependent controller for linear parameter-varying system is scheduled either via a *switching* scheme, e.g. Giua, et al. [33], or by *interpolating* among the local point designs, e.g. Kumar [44].

Robust control techniques such as H_2 or H_∞ control have recently become a popular concept in control of linear parameter-varying systems with un-modeled dynamics or unknown disturbances, e.g. Bruzelius [9], Fujiwara and Adachi [27], Gaspar, et al. [29], Wang and Tomizuka [89, 90] and Wu [93]. These techniques involve the solution of linear matrix inequalities and result in a constant state-feedback matrix ensuring that the transfer function from excitations to controlled outputs is lower than a prescribed small value, Gahinet, et al. [28]. The set of admissible parameter values can be treated in a direct

manner. In addition, bounds on the rates of change of the parameters can be incorporated to obtain a less conservative controller, e.g. Wang and Tomizuka [88]. The resulting controller has a stability and performance guarantee in the pre-defined operation region. However, a potential problem with these methods is the lack of performance.

Another approach for designing gain-scheduling control is the so-called *simultaneous Γ-stabilization method* presented by Ackermann [1] and Wang, et al. [91]. This technique permits the designer to specify a set of desired regions, joint or disjoint, in the complex root plane. Then a numerical algorithm is used to find the control parameters such that all the roots of the closed-loop systems resulting from the linearized plant models are within the specified regions. Although the performances of the closed-loop system can be improved by changing the desired regions in the complex plane, the simultaneous Γ-stabilization method is only suitable to controllers with a few components.

Petersen, et al. [58] use the *constrained LQR method* to design gain-scheduling for a wheel-slip-control model, resulting in a parameter-dependent controller scheduled by the car velocity. Good performance and robustness of the model are shown through analysis and experimental results. However, this approach is limited within a specific operation region and requires special experiences for designing the weighting matrices.

1.2 Outline of the Dissertation

Following this introduction chapter, the remainder of the thesis is divided into six chapters. Chapter 2 describes the fundamentals of multibody system dynamics. The equations of motion of multibody systems are established based on analyzing the kinematics and kinetics. Additionally, reduced and linearized forms of the equations of motion are presented which will be used for control analysis.

In Chapter 3 a three-degree-of-freedom spatial car model for studying the vehicle's lateral dynamics is introduced. To define the equations of motion, a plane track model describing yaw motion of the car is presented. The linearized equations of motion and their state-space representation are then introduced. Discussions on special cases of the

general spatial car model result in a simplified model to be used for optimal control analysis. For simulation, a spatial car model is built in MATLAB/Simulink.

An optimal control law for the spatial car model is defined in Chapter 4 based on the linear quadratic regulator (LQR) control. The LQR problem is shown first for linear systems without disturbances, which results in an optimal state-feedback controller, and then extended to linear systems with measurable disturbances, which leads to an optimal disturbance-feed forward controller. Automotive performance criteria specified for the spatial car model are also introduced in this chapter. The effectiveness of active suspensions with LQR control compared to passive suspensions is shown based on simulation results for the spatial car model.

Some background information on multi-criterion optimization (MCO) is first presented in Chapter 5. Then, formulations of the compromise method and recursive knee approach are given in more detail. MCO problems for both passive and active suspension cases are defined. In order to reduce the number of design variables for the case of active suspension, an optimization procedure combining the MCO method with the LQR algorithm is proposed. The advantages as well as drawbacks of the compromise method compared to the recursive knee approach for finding the Pareto frontier are discussed based on optimization results.

Chapter 6 introduces the method of designing gain-scheduling control for the linear parameter-varying spatial car model. First the operation region of the model is determined, considering the effects of the deformation of suspension and tires on the vehicle stability in cornering situations. Then, based on the local optimal controllers defined for specified operation points, a parameter-dependent controller is formulated that is able to vary continuously its parameters according to the changes of the system's varying parameters. To demonstrate the effectiveness of the designed parameter-dependent controller, vehicle handling test simulations are performed with input parameters obtained from the path generation problem defined for double-lane-change maneuvers.

Finally, conclusions and recommendations on future research are summarized in Chapter 7. Appendices provide the parameters of the studied car, the NEWEUL output file for the spatial car model and MATLAB.m-files used for the various investigations in this dissertation.

Chapter 2

Multibody System Dynamics

Many mechanical and structural systems such as vehicles, robots, mechanisms, and aircrafts consist of interconnected components that undergo large translational and rotational displacements and can be modeled as *multibody systems.* In this chapter, the *kinematics* and *kinetics* of multibody systems are formulated. Subsequently, the *equations of motion* of multibody systems in both nonlinear and linearized form are presented.

2.1 Multibody Systems

In general, a *multibody system* is defined to be a finite set of elements such as rigid bodies and/or particles, bearings, joints and supports, springs and dampers, active force and/or position actuators as illustrated in Figure 2.1 and Figure 2.2. For the mathematical description of these elements, the following assumptions are agreed upon, Schiehlen [70]:

1. A multibody system consists of rigid bodies and ideal joints. A body may degenerate to a particle or to a body without inertia. The ideal joints include the rigid joint, the joint with completely prescribed motion (rheonomic constraint) and the vanishing joint (free motion).
2. The topology of the multibody system is arbitrary; chains, trees and closed loops are admitted.
3. Joints and actuators are summarized in open libraries of standard elements.
4. Subsystems may be added to existing components of the multibody system.

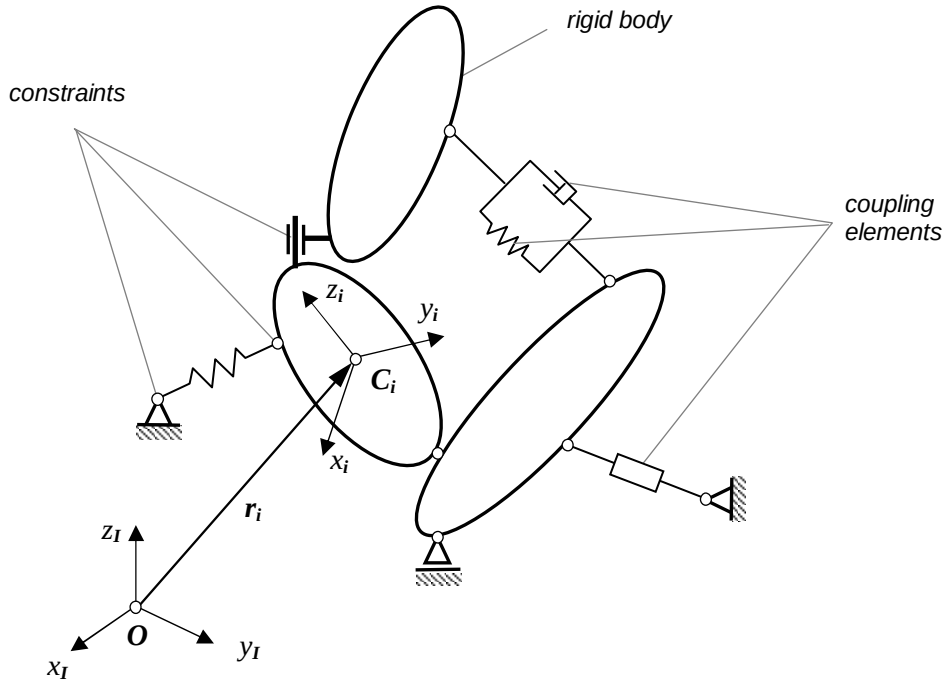

Figure 2.1: Multibody system

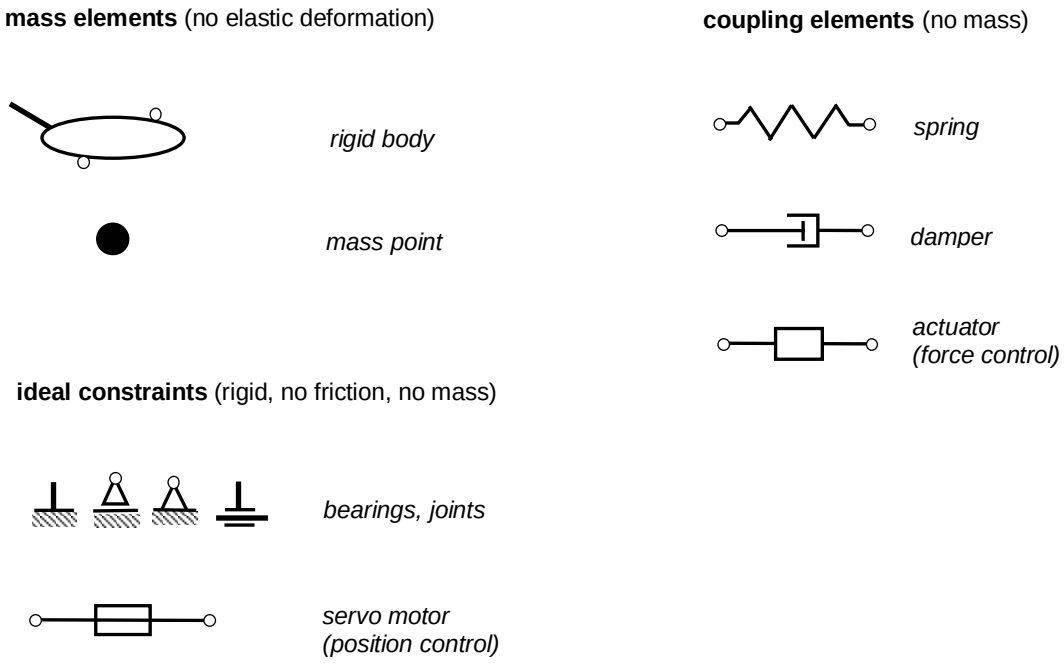

Figure 2.2: Elements of multibody systems (and idealizations)

The topological structure of a multibody system can be possibly tree structure or system with closed kinematical loops. The most commonly mentioned classification of constraints is scleronomic vs. rheonomic according to their time variation characteristic or holonomic vs. non-holonomic according to the constraint motion type. More detailed descriptions about multibody systems can be found in Bestle [7], Popp and Schiehlen [59] and Schiehlen [69].

For dynamical analysis, the multibody system has to be described mathematically by equations of motions. In the following sections the general theory for holonomic and non-holonomic systems will be presented using a minimal number of generalized coordinates for a unique representation of the motion.

2.2 Kinematics of Multibody Systems

There are basically two approaches in choosing coordinates to describe the kinematics of multibody systems, generalized, i.e. independent, coordinates and dependent coordinates. The former one leads to a kinematics description in minimal form, whereas the later one results in the descriptor form. Multibody systems with chain or tree structure can always be described with generalized coordinates and subsequently by ordinary differential equations (ODEs). Multibody systems with closed loops on the contrary cannot be always described with independent coordinates. The introduction of additional dependent coordinates in this case requires additional algebraic constraint equations resulting in a coupled differential-algebraic system of equations of motion (DAE).

The degrees of freedom (DoFs) f of a spatial multibody system with p bodies and q independent constraints can be calculated as $f = 6p - q$. Accordingly f generalized coordinates $\boldsymbol{y} = [\, y_1, y_2, \ldots, y_f \,]^T$ can be chosen to describe the translational and rotational motion of each body K_i, $i = 1(1)\, p$. The translation can be described with the position vector $\boldsymbol{r}_i$ of the center of gravity (CG), whereas the orientation may be described by a matrix of directional cosines $\boldsymbol{S}_i$. In an inertial reference frame, they can be described as functions of the generalized coordinates of following form:

$$\boldsymbol{r}_i = \boldsymbol{r}_i(\boldsymbol{y}, t),$$

$$\boldsymbol{S}_i = \boldsymbol{S}_i(\boldsymbol{y}, t), \qquad i = 1(1)\, p. \tag{2.1}$$

Through total differentiation with respect to time, the translational velocity $\boldsymbol{v_i}$ and angular velocity $\boldsymbol{\omega_i}$ of each body using the infinitesimal 3x1 vector of rotation $\boldsymbol{ds_i}$ can be expressed as

$$\boldsymbol{v_i} = \frac{d\boldsymbol{r_i}}{dt} = \frac{\partial \boldsymbol{r_i}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}\dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{r_i}(\boldsymbol{y},t)}{\partial t} =: \boldsymbol{J_{Ti}}(\boldsymbol{y},t)\dot{\boldsymbol{y}} + \bar{\boldsymbol{v}}_i(\boldsymbol{y},t),$$

$$\boldsymbol{\omega_i} = \frac{d\boldsymbol{s_i}}{dt} = \frac{\partial \boldsymbol{s_i}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}\dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{s_i}(\boldsymbol{y},t)}{\partial t} =: \boldsymbol{J_{Ri}}(\boldsymbol{y},t)\dot{\boldsymbol{y}} + \bar{\boldsymbol{\omega}}_i(\boldsymbol{y},t). \qquad (2.2)$$

The Jacobi matrices $\boldsymbol{J_{Ti}}$ and $\boldsymbol{J_{Ri}}$ of translational and angular velocity characterize the mapping from generalized to physical velocity space. These Jacobi matrices are necessary for the later application of d'Alembert's principle to eliminate the constraint reactions. The second term in Equations (2.2) will only occur with rheonomic constraints, they present the local velocity independent of $\dot{\boldsymbol{y}}$. Likely, the translational and angular accelerations $\boldsymbol{a_i}$ and $\boldsymbol{\alpha_i}$ can be calculated through repeated total differentiation:

$$\boldsymbol{a_i} = \frac{d\boldsymbol{v_i}}{dt} = \frac{\partial \boldsymbol{v_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial \dot{\boldsymbol{y}}}\ddot{\boldsymbol{y}} + \frac{\partial \boldsymbol{v_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial \boldsymbol{y}}\dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{v_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial t} =: \boldsymbol{J_{Ti}}(\boldsymbol{y},t)\ddot{\boldsymbol{y}} + \bar{\boldsymbol{a}}_i(\boldsymbol{y},\dot{\boldsymbol{y}},t),$$

$$\boldsymbol{\alpha_i} = \frac{d\boldsymbol{\omega_i}}{dt} = \frac{\partial \boldsymbol{\omega_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial \dot{\boldsymbol{y}}}\ddot{\boldsymbol{y}} + \frac{\partial \boldsymbol{\omega_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial \boldsymbol{y}}\dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{\omega_i}(\dot{\boldsymbol{y}},\boldsymbol{y},t)}{\partial t} =: \boldsymbol{J_{Ri}}(\boldsymbol{y},t)\ddot{\boldsymbol{y}} + \bar{\boldsymbol{\alpha}}_i(\boldsymbol{y},\dot{\boldsymbol{y}},t). \qquad (2.3)$$

The 3x1 vector $\bar{\boldsymbol{a}}_i$ of local translational acceleration and $\bar{\boldsymbol{\alpha}}_i$ of local angular acceleration contain the $\ddot{\boldsymbol{y}}$ independent acceleration terms.

2.3 Kinetics of Multibody Systems (Newton-Euler Equations)

The main purpose of the dynamic equations of multibody systems is to find a connection between motion and the acting forces. Basic approaches to the dynamics of multibody systems are distinguished as synthetic (vector) and analytic (scalar) approaches. The Newton-Euler formalism introduced here is essentially a synthetic approach.

For application of Newton's and Euler's law requires separation of the constrained body K_i from its interacting bodies by replacing the ideal constraints with equivalent constraint reactions and coupling elements by applied forces. Newton's equations of motion and Euler's dynamic equations can then be formulated as

$$m_i \boldsymbol{a}_i = \boldsymbol{f}_i^a + \boldsymbol{f}_i^r,$$

$$\boldsymbol{I}_i \boldsymbol{\alpha}_i + \tilde{\boldsymbol{\omega}}_i \boldsymbol{I}_i \boldsymbol{\omega}_i = \boldsymbol{l}_i^a + \boldsymbol{l}_i^r, \qquad i = 1(1)p. \tag{2.4}$$

In these equations, the mass property of the rigid body K_i is represented by its mass m_i and the 3x3 inertia tensor $\boldsymbol{I}_i$ relative to its center of gravity $\boldsymbol{C}_i$. The forces acting on the rigid body and the moment relative to its center of gravity are divided into applied forces $\boldsymbol{f}_i^a$ and moments $\boldsymbol{l}_i^a$, and reaction forces $\boldsymbol{f}_i^r$ and moments $\boldsymbol{l}_i^r$. The skew-symmetric tensor $\tilde{\boldsymbol{\omega}}_i$ is defined as

$$\tilde{\boldsymbol{\omega}}_i = \dot{\boldsymbol{S}}_i \boldsymbol{S}_i^T = \begin{bmatrix} 0 & -\omega_{zi} & \omega_{yi} \\ \omega_{zi} & 0 & -\omega_{xi} \\ -\omega_{yi} & \omega_{xi} & 0 \end{bmatrix}, \qquad i = 1(1)p. \tag{2.5}$$

Equation (2.4) consist of totally 6p Newton-Euler equations of motion for a multibody system with only f DoFs for both the f variables $\boldsymbol{y}$ and the reactions. With vector variables:

$$\boldsymbol{q}_{Ti}^c := m_i \bar{\boldsymbol{a}}_i,$$

$$\boldsymbol{q}_{Ri}^c := \boldsymbol{I}_i \bar{\boldsymbol{\alpha}}_i + \tilde{\boldsymbol{\omega}}_i \boldsymbol{I}_i \boldsymbol{\omega}_i, \qquad i = 1(1)p, \tag{2.6}$$

representing gyroscopic, Coriolis and centrifugal forces, Equations (2.4) may be rewritten as

$$m_i \boldsymbol{J}_{Ti} \ddot{\boldsymbol{y}} + \boldsymbol{q}_{Ti}^c = \boldsymbol{f}_i^a + \boldsymbol{f}_i^r,$$

$$\boldsymbol{I}_i \boldsymbol{J}_{Ri} \ddot{\boldsymbol{y}} + \boldsymbol{q}_{Ri}^c = \boldsymbol{l}_i^a + \boldsymbol{l}_i^r, \qquad i = 1(1)p. \tag{2.7}$$

The reaction forces and moments in (2.7) can be further expressed in qx1 general constraint forces $\boldsymbol{g} = [\, g_1, g_2, \ldots, g_q \,]^T$, with the translational and rotational distribution matrices $\boldsymbol{F}_i$ and $\boldsymbol{L}_i$:

$$\boldsymbol{f}_i^r = \boldsymbol{F}_i(\boldsymbol{y}, t)\, \boldsymbol{g},$$

$$\boldsymbol{l}_i^r = \boldsymbol{L}_i(\boldsymbol{y}, t)\, \boldsymbol{g}, \qquad i = 1(1)p. \tag{2.8}$$

Equation (2.7) can be summarized as Newton-Euler equations

$$\overline{\overline{\boldsymbol{M}}}\ \overline{\boldsymbol{J}}\,\ddot{\boldsymbol{y}} + \overline{\boldsymbol{q}}^{c} = \overline{\boldsymbol{q}}^{a} + \overline{\boldsymbol{Q}}\,\boldsymbol{g} \tag{2.9}$$

by introducing the following global matrices and vectors, respectively: 6*p*x6*p* global mass matrix $\overline{\overline{\boldsymbol{M}}}$, 6*p*x*f* global Jacobi matrix $\overline{\boldsymbol{J}}$, 6*p*x1 global vector of applied forces $\overline{\boldsymbol{q}}^{a}$, 6*p*x1 global vector of gyroscopic, Coriolis and centrifugal forces $\overline{\boldsymbol{q}}^{c}$, as well as the 6*p*x*q* global distribution matrix of reaction forces $\overline{\boldsymbol{Q}}$ as:

$$\begin{aligned}
\overline{\overline{\boldsymbol{M}}} &= diag\left\{m_1\boldsymbol{I},\ ...,\ m_p\boldsymbol{I},\ \boldsymbol{I}_1,\ ...,\ \boldsymbol{I}_p\right\},\\
\overline{\boldsymbol{J}} &= \left[\boldsymbol{J}_{T1}^{T},\ ...,\ \boldsymbol{J}_{Tp}^{T}\ \middle|\ \boldsymbol{J}_{R1}^{T},\ ...,\ \boldsymbol{J}_{Rp}^{T}\right]^{T},\\
\overline{\boldsymbol{q}}^{a} &= \left[\boldsymbol{f}_1^{aT},\ ...,\ \boldsymbol{f}_p^{aT}\ \middle|\ \boldsymbol{l}_1^{aT},\ ...,\ \boldsymbol{l}_p^{aT}\right]^{T},\\
\overline{\boldsymbol{q}}^{c} &= \left[\boldsymbol{q}_{T1}^{c\ T},\ ...,\ \boldsymbol{q}_{Tp}^{c\ T}\ \middle|\ \boldsymbol{q}_{R1}^{c\ T},\ ...,\ \boldsymbol{q}_{Rp}^{c\ T}\right]^{T},\\
\overline{\boldsymbol{Q}} &= \left[\boldsymbol{F}_1^{T},\ ...\ \boldsymbol{F}_p^{T}\ \middle|\ \boldsymbol{L}_1^{T},\ ...,\ \boldsymbol{L}_p^{T}\right]^{T}
\end{aligned} \tag{2.10}$$

where $\boldsymbol{I}$ denotes the 3x3-identity matrix.

2.4 Reduction and Linearization of the Equations of Motion

According to d'Alembert's principle, the virtual work of reaction forces vanishes for all motions which are consistent with the constraints. This can be expressed by an orthogonal relationship between the global Jacobi matrix and the global distribution matrix of reaction forces, Schiehlen [69]:

$$\overline{\boldsymbol{J}}^{T}\,\overline{\boldsymbol{Q}} = \boldsymbol{0}. \tag{2.11}$$

By multiplication of Equation (2.9) with the transposed global Jacobi matrix from the left, the reaction forces $\boldsymbol{g}$ can be eliminated as follows:

$$\underbrace{\overline{\boldsymbol{J}}^{T}\,\overline{\overline{\boldsymbol{M}}}\,\overline{\boldsymbol{J}}}_{:=\boldsymbol{M}}\ddot{\boldsymbol{y}} + \underbrace{\overline{\boldsymbol{J}}^{T}\,\overline{\boldsymbol{q}}^{c}}_{:=\boldsymbol{k}} = \underbrace{\overline{\boldsymbol{J}}^{T}\,\overline{\boldsymbol{q}}^{a}}_{:=\boldsymbol{q}} + \underbrace{\overline{\boldsymbol{J}}^{T}\,\overline{\boldsymbol{Q}}}_{\boldsymbol{0}}\boldsymbol{g}, \tag{2.12}$$

and the equations of motion expressed in general coordinates can be derived as:

$$\boldsymbol{M}(\boldsymbol{y},t)\ddot{\boldsymbol{y}} + \boldsymbol{k}(\boldsymbol{y},\dot{\boldsymbol{y}},t) = \boldsymbol{q}(\boldsymbol{y},\dot{\boldsymbol{y}},t). \tag{2.13}$$

These equations take on the form of f non-linear ODEs of 2nd order. The symmetric, positive definite fxf mass matrix $\boldsymbol{M}$, the fx1 vector of general gyroscopic, centrifugal and Coriolis forces $\boldsymbol{k}$, and the fx1 vector of general applied forces $\boldsymbol{q}$ are defined according to (2.12).

In many technical applications, vibrations with respect to prescribed motions or equilibrium positions will be small. Then the generalized coordinates $\boldsymbol{y}$ can be considered as small and the equations of motion can be linearized.

In case of holonomic multibody systems, the vibrations around the given motion, which is represented by the vector $\boldsymbol{y}_0(t)$, can be described as

$$\boldsymbol{y}(t) = \boldsymbol{y}_0(t) + \boldsymbol{\eta}(t) \tag{2.14}$$

where the fx1 position vector $\boldsymbol{\eta}(t)$ and the fx1 velocity vector $\dot{\boldsymbol{\eta}}(t)$ are always much smaller than some comparison vectors. Using the extension of Taylor's series and accepting the differentiable property of the vectors, the terms in (2.13) have the following linearized forms:

$$\begin{aligned}\boldsymbol{k}(\boldsymbol{y},\dot{\boldsymbol{y}},t) &= \underbrace{\boldsymbol{k}(\boldsymbol{y}_0(t),\dot{\boldsymbol{y}}_0(t),t)}_{:=\boldsymbol{k}_0(t)} + \underbrace{\left.\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}}\right|_{(\boldsymbol{y}_0,\dot{\boldsymbol{y}}_0)}}_{:=\boldsymbol{K}_{\boldsymbol{y}}(t)}\boldsymbol{\eta}(t) + \underbrace{\left.\frac{\partial \boldsymbol{k}}{\partial \dot{\boldsymbol{y}}}\right|_{(\boldsymbol{y}_0,\dot{\boldsymbol{y}}_0)}}_{:=\boldsymbol{K}_{\dot{\boldsymbol{y}}}(t)}\dot{\boldsymbol{\eta}}(t) + \ldots \\ &\approx \boldsymbol{k}_0(t) + \boldsymbol{K}_{\boldsymbol{y}}(t)\boldsymbol{\eta}(t) + \boldsymbol{K}_{\dot{\boldsymbol{y}}}(t)\dot{\boldsymbol{\eta}}(t),\end{aligned} \tag{2.15}$$

$$\begin{aligned}\boldsymbol{q}(\boldsymbol{y},\dot{\boldsymbol{y}},t) &= \underbrace{\boldsymbol{q}(\boldsymbol{y}_0(t),\dot{\boldsymbol{y}}_0(t),t)}_{:=\boldsymbol{q}_0(t)} + \underbrace{\left.\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{y}}\right|_{(\boldsymbol{y}_0,\dot{\boldsymbol{y}}_0)}}_{:=\boldsymbol{Q}_{\boldsymbol{y}}(t)}\boldsymbol{\eta}(t) + \underbrace{\left.\frac{\partial \boldsymbol{q}}{\partial \dot{\boldsymbol{y}}}\right|_{(\boldsymbol{y}_0,\dot{\boldsymbol{y}}_0)}}_{:=\boldsymbol{Q}_{\dot{\boldsymbol{y}}}(t)}\dot{\boldsymbol{\eta}}(t) + \ldots \\ &\approx \boldsymbol{q}_0(t) + \boldsymbol{Q}_{\boldsymbol{y}}(t)\boldsymbol{\eta}(t) + \boldsymbol{Q}_{\dot{\boldsymbol{y}}}(t)\dot{\boldsymbol{\eta}}(t).\end{aligned} \tag{2.16}$$

The first term in (2.13) can be written as:

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{y},t)\ddot{\boldsymbol{y}} &= \boldsymbol{M}(\boldsymbol{y},t)[\ddot{\boldsymbol{y}}_0(t)+\ddot{\boldsymbol{\eta}}(t)] \\ &= \underbrace{\boldsymbol{M}(\boldsymbol{y}_0(t),t)}_{:=\boldsymbol{M}_0(t)}[\ddot{\boldsymbol{y}}_0(t)+\ddot{\boldsymbol{\eta}}(t)] + \left.\frac{\partial\left(\boldsymbol{M}[\ddot{\boldsymbol{y}}_0(t)+\ddot{\boldsymbol{\eta}}(t)]\right)}{\partial \boldsymbol{y}}\right|_{(\boldsymbol{y}_0)}\boldsymbol{\eta}(t) + \ldots \\ &\approx \boldsymbol{M}_0(t)\ddot{\boldsymbol{y}}_0(t) + \boldsymbol{M}_0(t)\ddot{\boldsymbol{\eta}}(t) + \underbrace{\left.\frac{\partial\left(\boldsymbol{M}\ddot{\boldsymbol{y}}_0(t)\right)}{\partial \boldsymbol{y}}\right|_{(\boldsymbol{y}_0)}}_{:=\boldsymbol{M}_{\boldsymbol{y}}(t)}\boldsymbol{\eta}(t).\end{aligned} \tag{2.17}$$

Substituting (2.15), (2.16) and (2.17) for (2.13) yields the linearized equations of motion in the form:

$$\boldsymbol{M}(t)\ddot{\boldsymbol{y}} + \boldsymbol{P}(t)\dot{\boldsymbol{y}} + \boldsymbol{Q}(t)\boldsymbol{y} = \boldsymbol{h}(t) \tag{2.18}$$

where

$$\begin{aligned}
\boldsymbol{M}(t) &:= \boldsymbol{M}_0(t), \\
\boldsymbol{P}(t) &:= \boldsymbol{K}_{\dot{\boldsymbol{y}}}(t) - \boldsymbol{Q}_{\dot{\boldsymbol{y}}}(t), \\
\boldsymbol{Q}(t) &:= \boldsymbol{M}_{\boldsymbol{y}}(t) + \boldsymbol{K}_{\boldsymbol{y}}(t) - \boldsymbol{Q}_{\boldsymbol{y}}(t), \\
\boldsymbol{h}(t) &:= \boldsymbol{q}_0(t) - \boldsymbol{k}_0(t) - \boldsymbol{M}_0(t)\ddot{\boldsymbol{y}}_0(t).
\end{aligned} \tag{2.19}$$

If further $\boldsymbol{M}(t)$, $\boldsymbol{P}(t)$ and $\boldsymbol{Q}(t)$ are independent of time, the multibody systems can be treated as linear time-invariant (LTI) system described by

$$\boldsymbol{M}\ddot{\boldsymbol{y}} + (\boldsymbol{D}+\boldsymbol{G})\dot{\boldsymbol{y}} + (\boldsymbol{K}+\boldsymbol{N})\boldsymbol{y} = \boldsymbol{h}(t) \tag{2.20}$$

where the matrix of velocity-dependent forces $\boldsymbol{P}$ is split into a symmetric matrix of damping forces $\boldsymbol{D}$ and a skew-symmetric matrix of gyroscopic forces $\boldsymbol{G}$, respectively, and the matrix of position-dependent forces $\boldsymbol{Q}$ is split into matrices of stiffness $\boldsymbol{K} = \boldsymbol{K}^T$ and non-conservative forces $\boldsymbol{N} = -\boldsymbol{N}^T$, respectively. All matrices hereby are of dimension $f \times f$. Vector $\boldsymbol{h}(t)$ is an excitation vector of dimension $f \times 1$ representing control or disturbing input forces.

For non-holonomic systems the velocity degrees of freedom are reduced by non-holonomic constraints. With the application of Jordain's principle, the equations of motion in both nonlinear and linearized form can be obtained similarly. More details about non-holonomic systems can be found in Bestle [7] and Schiehlen [69].

Chapter 3

Passenger Car Modeling

In this chapter, a *spatial car model* for a vehicle with a double-control-arm suspension will be developed. The yaw motion of the car will be derived from a *plane track model*. The *linearized equations of motion* obtained from the computer-aided multibody system program NEWEUL will be transformed into the *state-space representation* form. Finally, a simplified spatial car model will be presented which will be used to design optimal control.

3.1 Suspension Forces

The influences of suspension geometry are often ignored in conventional quarter-car models. In this section, modified suspension parameters characterizing the effects of suspension geometry will be defined by comparing the virtual works generated by the forces acting on the car body of a double-control-arm suspension and those of a conventional quarter-car model. The virtual-work method introduced in this section can be applied analogously for other types of suspension to find properly modified suspension parameters.

3.1.1 Double-control-arm suspension

The schematic diagram of a double-control-arm suspension system is shown in Figure 3.1. In this model, the directions of the spring-damper and the actuator at the static equilibrium are described by angles φ_0 and φ_{C0} respectively, while that of the lower

control arm is presented by angle ξ_0. The model has two degrees of freedom, the vertical displacement of the sprung mass z_S and the displacement of the unsprung mass which may be represented by the rotational angle ξ of the lower suspension arm.

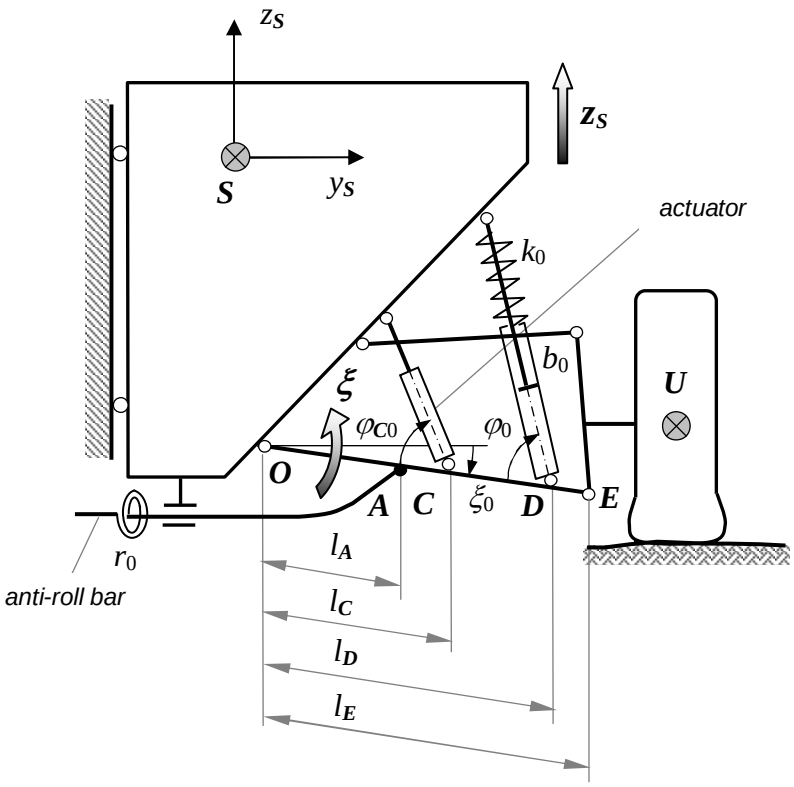

Figure 3.1: Plane model of double-control-arm suspension

The given parameters of the model are the stiffness of the spring k_0, the damping coefficient of the damper b_0, the rotational stiffness of the anti-roll bar r_0 and positions of joints. The suspension forces acting on the car body (sprung mass) result from spring, damper, actuator and anti-roll bar.

- **Spring and damping forces**

Figure 3.2 illustrates the definition of the spring force vector $\boldsymbol{F_K}$ and the damping force vector $\boldsymbol{F_B}$ when joint $\boldsymbol{D}$ connecting the spring-damper with the lower control arm moves to $\boldsymbol{D}$'. For small rotational angle ξ, i.e. $\xi << 1$, vector $\boldsymbol{\delta_D}$ representing the displacement of joint

D can be treated to be orthogonal to the lower control arm ***OD*** and its value can be defined by $\delta_D = l_D \sin(\xi) \approx l_D\,\xi$, resulting in the *dynamic deflection of the spring-damper*:

$$\Delta l \approx \delta_D \sin(\varphi_0) \approx l_D \xi \sin(\varphi_0). \tag{3.1}$$

The *spring force* F_K is proportional to the sum of dynamic deflection Δl and static deflection Δl_0 of the spring, i.e.

$$F_K = k_0(\Delta l + \Delta l_0), \tag{3.2}$$

or with (3.1):

$$F_K = k_0 l_D \xi \sin(\varphi_0) + k_0 \Delta l_0 . \tag{3.3}$$

The *damping force* F_B can be computed approximately by

$$F_B = b_0 \Delta \dot{l} = b_0 l_D \dot{\xi} \sin(\varphi_0). \tag{3.4}$$

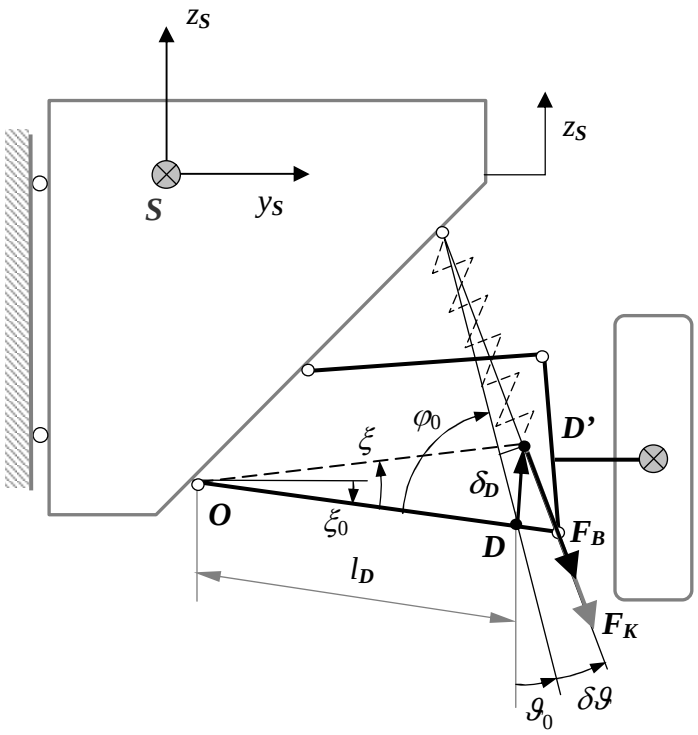

Figure 3. 2: Definition of the spring and damping forces

By defining an instantaneous velocity center ***P*** of the unsprung mass as shown in Figure 3.3, the rotational angle ξ and velocity $\dot{\xi}$ of the lower suspension arm can be expressed by

$$\xi \approx \frac{L_E}{l_E L_U}\ \Delta z \tag{3.5}$$

and

$$\dot{\xi} \approx \frac{L_E}{l_E L_U}\ \Delta \dot{z}\,, \tag{3.6}$$

respectively, where $\Delta z = (z_U - z_S)$ is the relative vertical displacement between the sprung and unsprung mass. Substituting ξ into Equation (3.3) yields

$$F_K = k_0\ \underbrace{l_D \frac{L_E}{l_E L_U} \sin(\varphi_0)}_{:=\lambda_D}\ \Delta z\ +\ k_0\ \Delta l_0\,. \tag{3.7}$$

With $\dot{\xi}$ defined by (3.6), the damping force F_B in (3.4) can be expressed by

$$F_B = b_0\ \lambda_D\ \Delta \dot{z}\,. \tag{3.8}$$

In the above, λ_D is the coefficient representing the influences of the suspension geometry on the spring and damping forces.

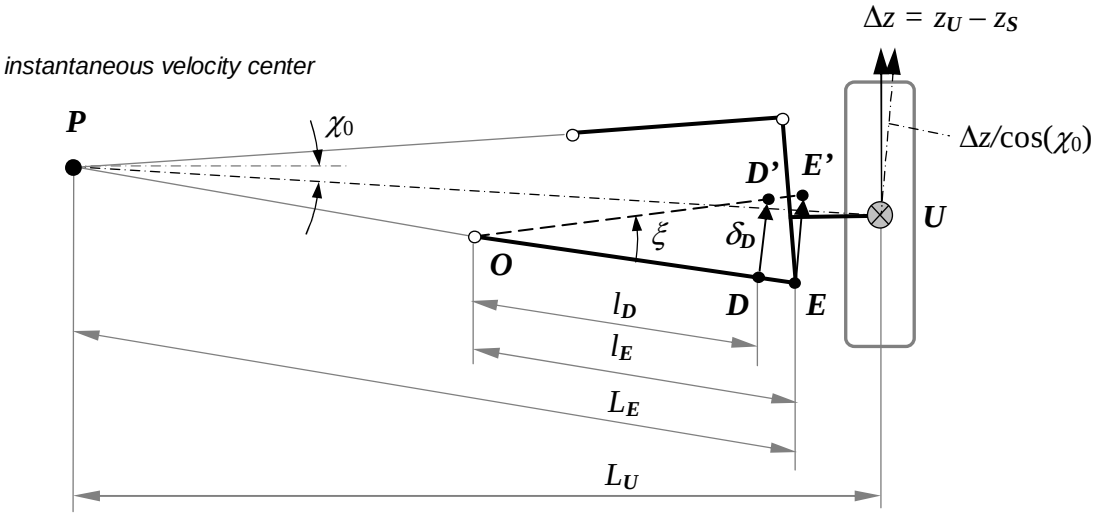

Figure 3. 3: Definition of the rotational angle of the lower suspension arm

By introducing rotational matrices $\boldsymbol{S}_{\xi_0}$, $\boldsymbol{S}_{\xi}$ and $\boldsymbol{S}_{(\vartheta_0+\delta\vartheta)}$ corresponding to angles ξ_0, ξ and $(\vartheta_0+\delta\vartheta)$ as

$$\boldsymbol{S}_{\xi_0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\xi_0) & \sin(\xi_0) \\ 0 & -\sin(\xi_0) & \cos(\xi_0) \end{bmatrix}, \tag{3.9}$$

$$\boldsymbol{S}_{\xi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\xi) & -\sin(\xi) \\ 0 & \sin(\xi) & \cos(\xi) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\xi \\ 0 & \xi & 1 \end{bmatrix}, \tag{3.10}$$

$$\boldsymbol{S}_{(\vartheta_0+\delta\vartheta)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta_0+\delta\vartheta) & -\sin(\vartheta_0+\delta\vartheta) \\ 0 & \sin(\vartheta_0+\delta\vartheta) & \cos(\vartheta_0+\delta\vartheta) \end{bmatrix} \tag{3.11}$$

where $\vartheta_0 := \frac{\pi}{2} - (\varphi_0 + \xi_0)$ and $\delta\vartheta << 1$, the directional unit vector $\boldsymbol{e_D}$ of $\boldsymbol{F_K}$ and $\boldsymbol{F_B}$ in the coordinate system $\boldsymbol{S}$ fixed to the sprung mass can be defined by

$$\boldsymbol{e_D} = \boldsymbol{S}_{(\vartheta_0+\delta\vartheta)} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ \sin(\vartheta_0+\delta\vartheta) \\ -\cos(\vartheta_0+\delta\vartheta) \end{bmatrix} \approx \begin{bmatrix} 0 \\ \sin(\vartheta_0)+\cos(\vartheta_0)\delta\vartheta \\ -\cos(\vartheta_0)+\sin(\vartheta_0)\delta\vartheta \end{bmatrix} \tag{3.12}$$

while vector $\boldsymbol{\delta_D}$ in the coordinate system $\boldsymbol{S}$ can be defined from vectors $\boldsymbol{r_{OD}}$ and $\boldsymbol{r_{OD'}}$ as

$$\boldsymbol{\delta_D} = \boldsymbol{r_{OD'}} - \boldsymbol{r_{OD}} \tag{3.13}$$

with

$$\boldsymbol{r_{OD'}} = \boldsymbol{S}_{\xi_0}\, \boldsymbol{S}_{\xi} \begin{bmatrix} 0 \\ l_D \\ 0 \end{bmatrix} = l_D \begin{bmatrix} 0 \\ \cos(\xi_0) + \xi \sin(\xi_0) \\ -\sin(\xi_0) + \xi \cos(\xi_0) \end{bmatrix} \tag{3.14}$$

and

$$\boldsymbol{r_{OD}} = \boldsymbol{S}_{\xi_0} \begin{bmatrix} 0 \\ l_D \\ 0 \end{bmatrix} = l_D \begin{bmatrix} 0 \\ \cos(\xi_0) \\ -\sin(\xi_0) \end{bmatrix} \tag{3.15}$$

resulting in

$$\boldsymbol{\delta_D} = l_{\boldsymbol{D}}\, \xi \begin{bmatrix} 0 \\ \sin(\xi_0) \\ \cos(\xi_0) \end{bmatrix}. \tag{3.16}$$

With the directional unit vector $\boldsymbol{e_D}$ (3.12), the vector of spring force $\boldsymbol{F_K}$ can be defined as

$$\boldsymbol{F_K} = F_{\boldsymbol{K}}\, \boldsymbol{e_D} = F_{\boldsymbol{K}} \begin{bmatrix} 0 \\ \sin(\vartheta_0) + \cos(\vartheta_0)\delta\vartheta \\ -\cos(\vartheta_0) + \sin(\vartheta_0)\delta\vartheta \end{bmatrix}$$

$$= F_{\boldsymbol{K}} \begin{bmatrix} 0 \\ \sin(\vartheta_0) \\ -\cos(\vartheta_0) \end{bmatrix} + F_{\boldsymbol{K}} \begin{bmatrix} 0 \\ \cos(\vartheta_0) \\ \sin(\vartheta_0) \end{bmatrix} \delta\vartheta\ . \tag{3.17}$$

The *virtual work* generated by the spring force can be computed by

$$\delta W_{\boldsymbol{K}} = \boldsymbol{F_K^T}\ \boldsymbol{\delta_D}\ . \tag{3.18}$$

Substituting (3.16) and (3.17) into (3.18) and taking into account $\xi, \delta\vartheta << 1$ yields

$$\delta W_{\boldsymbol{K}} = \boldsymbol{F_K^T}\ \boldsymbol{\delta_D} = F_{\boldsymbol{K}} l_{\boldsymbol{D}}\, \xi\ \left(-\cos(\vartheta_0 + \xi_0)\right)\ =\ F_{\boldsymbol{K}} l_{\boldsymbol{D}}\, \xi\ \left(-\cos\left(\frac{\pi}{2} - \varphi_0\right)\right)$$

$$=\ -F_{\boldsymbol{K}}\ l_{\boldsymbol{D}}\, \xi \sin(\varphi_0), \tag{3.19}$$

or with (3.5) and (3.7):

$$\delta W_{\boldsymbol{K}} = -F_{\boldsymbol{K}} \lambda_{\boldsymbol{D}}\ \Delta z = -\left(k_0 \lambda_{\boldsymbol{D}}^2\ \Delta z\ +\ k_0 \lambda_{\boldsymbol{D}}\ \Delta l_0\right)\Delta z. \tag{3.20}$$

Similarly, the virtual work generated by the damping force can be computed by

$$\delta W_{\boldsymbol{B}} = -b_0 \lambda_{\boldsymbol{D}}^2\ \Delta\dot{z}\ \Delta z\,. \tag{3.21}$$

The virtual works generated by the spring and damping forces of the double-control arm suspension model will be compared to those of a conventional quarter-car model to define the modified suspension stiffness and damping coefficient.

- **Actuator force**

The magnitude of the actuator force $\boldsymbol{F_C}$, denoted by u_0, is determined by control commands. By defining the directional unit vector of $\boldsymbol{F_C}$ and the displacement of joint $\boldsymbol{C}$ connecting the actuator with the lower suspension arm, we can obtain the virtual work generated by the actuator force as

$$\delta W_{\boldsymbol{C}} = -u_0 \underbrace{l_{\boldsymbol{C}} \frac{L_{\boldsymbol{E}}}{l_{\boldsymbol{E}} L_{\boldsymbol{U}}} \sin(\varphi_{\boldsymbol{C}0})}_{:=\lambda_{\boldsymbol{C}}} \Delta z \tag{3.22}$$

where $\lambda_{\boldsymbol{C}}$ represents the influences of the suspension geometry on the actuator forces.

- **Anti-roll force**

When the small vertical displacements of the left and the right wheels are different, the anti-roll bar with a rotational stiffness r_0 creates an anti-roll moment $\boldsymbol{M_U}$, see Figure 3.4, as

$$M_{\boldsymbol{U}} = r_0 \frac{(\delta_{\boldsymbol{Al}} - \delta_{\boldsymbol{Ar}})\cos(\xi_0)}{e} \tag{3.23}$$

where $\delta_{\boldsymbol{A}}$ is the displacement of joint $\boldsymbol{A}$ connecting anti-roll bar with lower suspension arm, the subscripts "*l*" and "*r*" denote the left and the right wheel of the car, respectively. This moment results in the anti-roll force $\boldsymbol{F_U}$ acting on the unsprung mass with the value:

$$F_{\boldsymbol{U}} = \frac{M_{\boldsymbol{U}}}{e} = r_0 \frac{(\delta_{\boldsymbol{Al}} - \delta_{\boldsymbol{Ar}})\cos(\xi_0)}{e^2} . \tag{3.24}$$

The virtual work generated by the anti-roll forces can be computed by

$$\delta W_{\boldsymbol{A}} = -F_{\boldsymbol{U}} (\delta_{\boldsymbol{Al}} - \delta_{\boldsymbol{Ar}})\cos(\xi_0), \tag{3.25}$$

or with (3.24):

$$\delta W_{\boldsymbol{A}} = -r_0 \left[\frac{(\delta_{\boldsymbol{Al}} - \delta_{\boldsymbol{Ar}})\cos(\xi_0)}{e} \right]^2 . \tag{3.26}$$

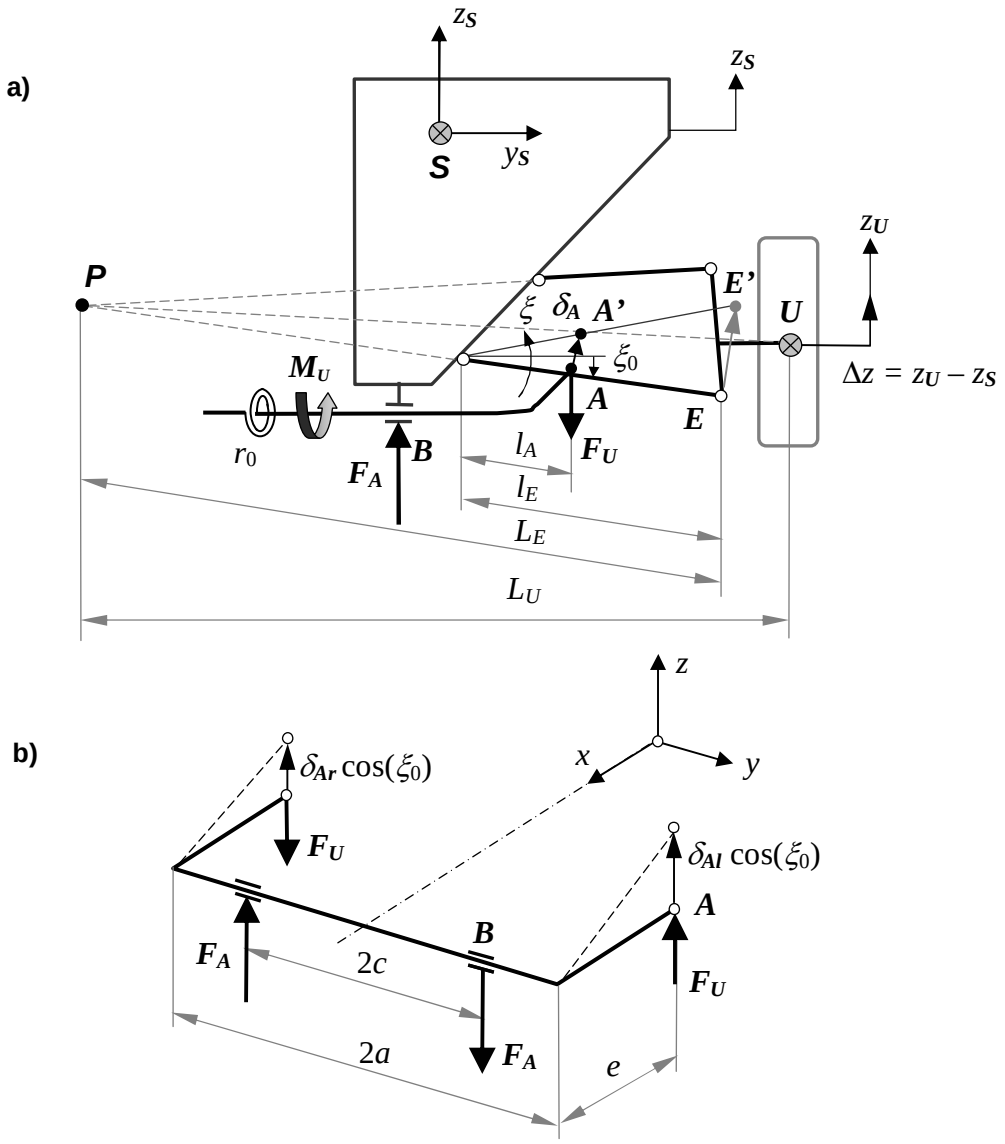

Figure 3.4: Definition of the anti-roll force

For small rotational angle ξ of the lower control arm, the displacement δ_A can be computed from Figure 3.4a as

$$\delta_A \approx l_A \xi , \tag{3.27}$$

or with ξ given in (3.5):

$$\delta_A \approx l_A \frac{L_E}{l_E L_U} \Delta z . \tag{3.28}$$

Substituting the above equation into (3.24) and (3.26) yields

$$F_U = r_0 \frac{1}{e^2} l_A \frac{L_E}{l_E L_U} \cos(\xi_0)(\Delta z_l - \Delta z_r), \tag{3.29}$$

$$\delta W_A = -r_0 \underbrace{\left[\frac{1}{e} l_A \frac{L_E}{l_E L_U} \cos(\xi_0) \right]^2}_{:= \lambda_A} (\Delta z_l - \Delta z_r)^2 \tag{3.30}$$

where λ_A is the coefficient representing the influences of the suspension geometry on the anti-roll forces. At the sprung mass, the value of the anti-roll force F_A is defined by the equilibrium condition

$$2 F_U a - 2 F_A c = 0 \tag{3.31}$$

resulting in

$$F_A = F_U \frac{a}{c} = r_0 \lambda_A \frac{a}{e c}. \tag{3.32}$$

3.1.2 Modified suspension parameters

Let us consider the conventional quarter car model illustrated in Figure 1.3 with spring stiffness *k*, the damping coefficient *b* and the value *u* of the actuator force F_C. For this model, the values of the spring force F_K with pre-stress $\hat{F}_{K0}$ and damping force F_B are defined by

$$F_K = k(z_U - z_S) + \hat{F}_{K_0} := k \Delta z + \hat{F}_{K_0}, \tag{3.33}$$

$$F_B = b(\dot{z}_U - \dot{z}_S) := b \Delta \dot{z} \tag{3.34}$$

where z_U and z_S are vertical displacements of unsprung and sprung mass, respectively.

Since the directions of the forces acting on the bodies and displacements are opposite, the virtual works resulting from the spring, damper and control force can be computed as

$$\delta W_K = -F_K \Delta z = -\left(k \Delta z + \hat{F}_{K_0}\right)\Delta z, \tag{3.35}$$

$$\delta W_B = -F_B \Delta z = -b \Delta \dot{z} \Delta z, \tag{3.36}$$

$$\delta W_C = -u \Delta z. \tag{3.37}$$

Comparing the above virtual works to those of the double-control-arm model defined by (3.20), (3.21) and (3.22), respectively, yields

$$k = k_0 \lambda_D^2 , \qquad \hat{F}_{K_0} = k_0 \lambda_D \Delta l_0 , \tag{3.38}$$

$$b = b_0 \lambda_D^2 , \tag{3.39}$$

$$u = u_0 \lambda_C . \tag{3.40}$$

For a half-car or full-car model constructed from a combination of quarter car models, the value of the anti-roll force $\boldsymbol{F_A}$ at the left and the right wheel can be computed by

$$F_A = r \left(\Delta z_l - \Delta z_r \right) \tag{3.41}$$

resulting in the virtual work

$$\delta W_A = - F_A \left(\Delta z_l - \Delta z_r \right) = - r \left(\Delta z_l - \Delta z_r \right)^2 \tag{3.42}$$

where r is the modified rotational stiffness of the anti-roll bar. Comparing to the virtual work (3.30) yields the anti-roll stiffness for a simplified car model:

$$r = r_0 \lambda_A^2 . \tag{3.43}$$

It should be noted that the unit of r is [*N*/*m*/*rad*] instead of [*Nm*/*rad*] due to the unit of λ_A (3.30).

With the modified suspension parameters defined by (3.38), (3.39), (3.40) and (3.43), the influences of the geometry of a double-control-arm suspension can be involved in the conventional simplified car models.

3.2 Three Degree-of-Freedom Spatial Car Model

Capturing all vehicle dynamic problems with one universal model can be quite difficult. Although including more number of elements in the model may increase the model's accuracy, it substantially increases the computation time. In order to study the influence of suspension characteristics on vehicle handling and stability, i.e. the lateral dynamics, a novel spatial car model is proposed with the following simplifications, Figure 3.5:

- the four wheels are treated as massless points that keep their traces along the road surface; the mass of suspensions is also ignored;
- the car body, i.e. the sprung mass, is considered as single rigid body that can rotate along its fixed roll and pitch axes. The pitch axis is assumed to go through the center of gravity of the car body.

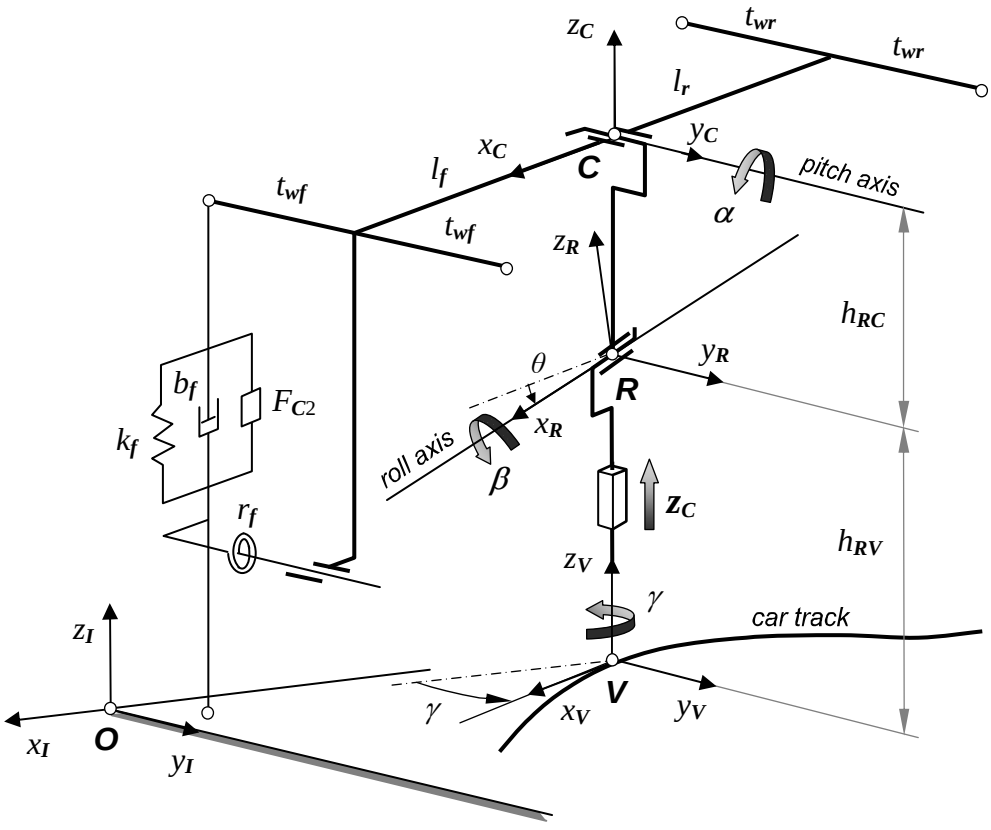

Figure 3.5: Three degree-of-freedom spatial car model in reference configuration

The *spatial car model* has three degrees of freedom: 1) the vertical motion expressed by z_C, 2) the rotational motion β about the roll axis which is inclined by a constant angle θ with respect to the horizontal axis, and 3) the rotational motion about the pitch axis denoted by pitch angle α. In order to describe the motion of the car body, three coordinate systems are introduced additionally to the absolute inertial reference frame $\{\boldsymbol{O}, x_I, y_I, z_I\}$, i.e. the track coordinate system $\{\boldsymbol{V}, x_V, y_V, z_V\}$, the car body roll motion coordinate system $\{\boldsymbol{R}, x_R, y_R, z_R\}$ and the car body fixed coordinate system $\{\boldsymbol{C}, x_C, y_C, z_C\}$. The direction of coordinate systems is defined according to ISO 8855, i.e. the positive *x*-axis points straight forward, the *y*-axis points to the left and the *z*-axis points upwards.

The trace of the chassis, i.e. the un-sprung mass including the four wheels, in the *x*-*y* plane of the reference frame $\boldsymbol{O}$ can be described with a coordinate system that translates only within the *x*-*y* plane and rotates only along the z_I-axis. This coordinate system is referred to as the track coordinate system $\boldsymbol{V}$. At equilibrium of the car, the *z*-axis of $\boldsymbol{V}$ runs through the center of gravity of the car body. It is obvious that the rotation of $\boldsymbol{V}$ represents the *yaw motion* of the car denoted by γ.

The roll motion coordinate system $\boldsymbol{R}$ is assumed to keep its origin directly above the coordinate system $\boldsymbol{V}$, i.e. $\boldsymbol{R}$ shifts only along the *z*-axis of $\boldsymbol{V}$. This shifting is indicated by z_C and it is one of the three degrees of freedom of the car body. Orientation of system $\boldsymbol{R}$ can be described by two consequent elementary rotations. The first one is a rotation about the *y*-axis of coordinate system $\boldsymbol{V}$ with a fixed angle θ defining the *roll axis* of the sprung mass. The consequent rotation is about the *x*-axis of frame $\boldsymbol{R}$ with *roll angle* β which along with the rotation axis describes the *roll motion* of the sprung mass and yields the second degree of freedom of the car body.

The last coordinate system $\boldsymbol{C}$ is fixed to the car body with its origin fixed to coordinate system $\boldsymbol{R}$ and a rotational degree about the *y*-axis. This rotation represents the *pitch motion* of the car and is characterized by angle α, which is the third degree of freedom of the car body.

Applied forces and moments on the car body result from springs, dampers and actuators of the four suspensions and anti-roll bars in the front and at the rear side of the car.

3.3 Plane Track Model

The car is assumed to move along a given trajectory and to keep its yaw orientation tangential to the track all the time. In order to describe the motion of the car, the track must be modeled first. A relatively simple and easy way to produce a car track is the division of the whole track path into sections. It is possible to reproduce all road courses which occur in real road systems using only three different path forms: straight-line, spiral and circular-arc segments as illustrated in Figure 3.6.

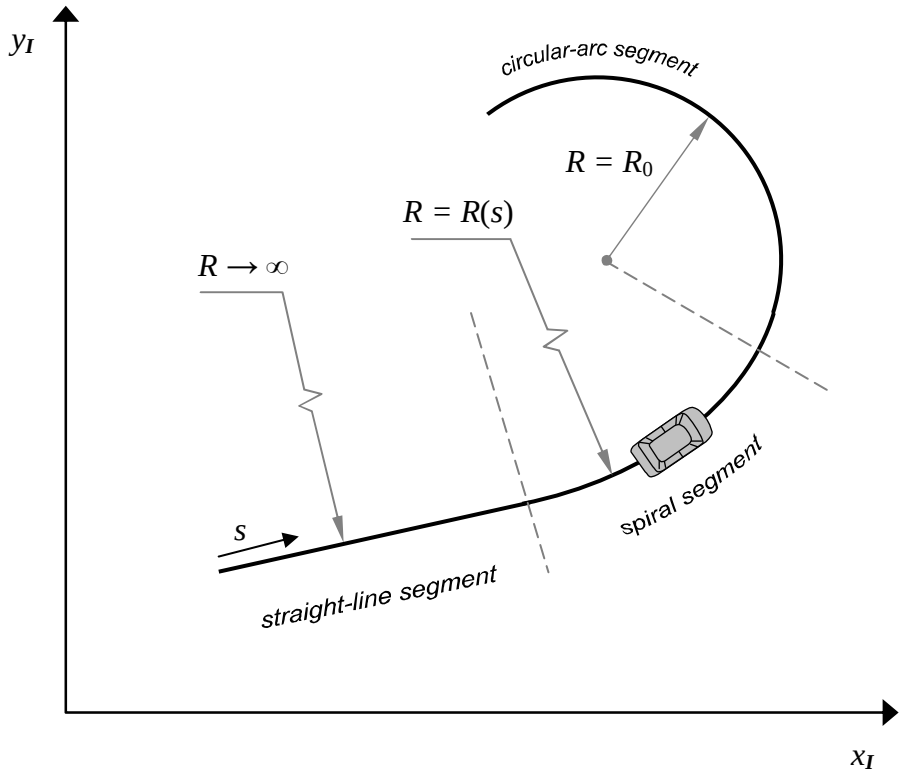

Figure 3.6: Basic path forms of the track

The track is described by a track coordinate s along the track and its curvature $\kappa = \kappa(s)$ which is the inverse of the curve radius $R(s)$, i.e. $|\kappa(s)| = 1/R(s)$, where $\kappa(s) > 0$ for a left curve and $\kappa(s) < 0$ for a right curve. The curvature of the track segments can be defined as follows:

- for straight-line segments: $R(s) \to \infty \;\Rightarrow\; \kappa(s) = 0,$ (3.44)
- for circular-arc segments: $R(s) = R_0 \;\Rightarrow\; \kappa(s) = 1/R_0 = const.,$ (3.45)
- for spiral segments running from s_0 to s_1:

  - from a straight line into a circular-arc e.g.:

$$R(s):\ \infty \to R_0 \Rightarrow \kappa(s) = \frac{1}{R_0}\left(\frac{s - s_0}{s_1 - s_0}\right), \tag{3.46}$$

  - from a circular-arc to a straight line:

$$R(s):\ R_0 \to \infty \Rightarrow \kappa(s) = \frac{1}{R_0}\left(\frac{s_1 - s}{s_1 - s_0}\right). \tag{3.47}$$

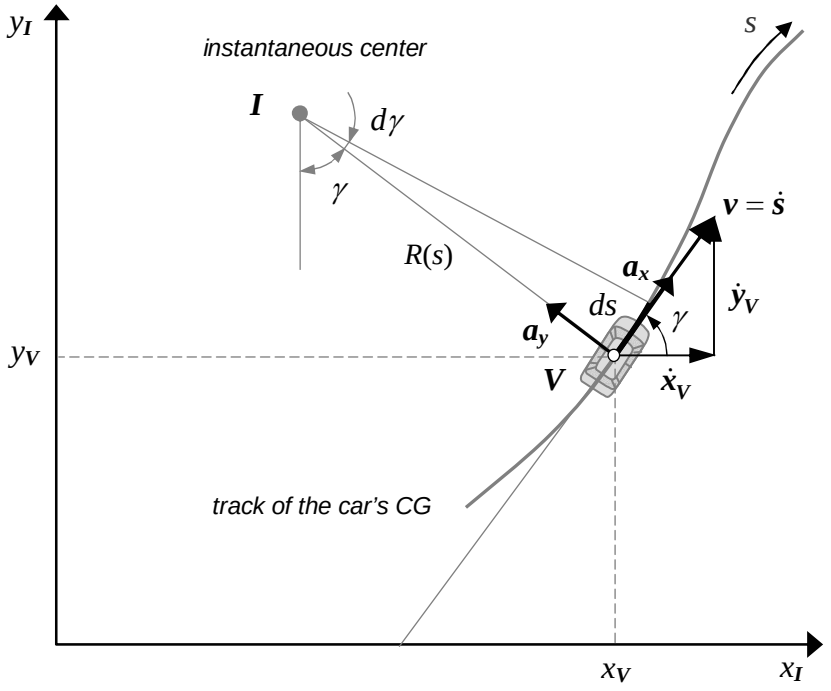

Figure 3.7: Plane track model

Figure 3.7 shows the plane track model used to describe the yaw motion of the car. As illustrated in the figure, the angular yaw velocity can be defined as following:

$$\dot{\gamma} = \frac{d\gamma}{dt} = \frac{1}{R(s)}\frac{ds}{dt} = \frac{1}{R(s)}\dot{s} = \kappa(s)v\,. \tag{3.48}$$

The trajectory of the track in the *x*-*y* plane of the inertial coordinate system may be computed from

$$\begin{aligned} \dot{x}_V &= v\cos(\gamma), \\ \dot{y}_V &= v\,\sin(\gamma). \end{aligned} \tag{3.49}$$

The accelerations $\ddot{x}_V$ and $\ddot{y}_V$ can be projected along tangent and normal directions of the moving frame $\boldsymbol{V}$ as

$$\begin{aligned} a_x &= \ \ \ddot{x}_V\cos(\gamma) + \ddot{y}_V\sin(\gamma), \\ a_y &= -\ddot{x}_V\sin(\gamma) + \ddot{y}_V\cos(\gamma). \end{aligned} \tag{3.50}$$

Alternatively the car longitudinal and lateral acceleration can be calculated from longitudinal and angular velocity as

$$\begin{aligned} a_x &= \dot{v}, \\ a_y &= \frac{v^2}{R(s)} = v\,[\,v\,\kappa(s)] = v\dot{\gamma}\,. \end{aligned} \tag{3.51}$$

3.4 Linearized Equations of Motion of the Spatial Car Model

In this section, the linearized equations of motion of the spatial car model will be defined which will be used for simulation and control analysis. As indicated in section 3.2, the vector of generalized coordinates for the spatial car model may be chosen as

$$\boldsymbol{y} := [\,z_C\,,\ \beta\,,\ \alpha\,]^{\boldsymbol{T}} \tag{3.52}$$

and the vector of general applied forces and moments resulting from the suspension and expressed in the car body fixed frame may be summarized as

$$\boldsymbol{q^a} := [\,fz_C\,,\ l_\beta\,,\ l_\alpha\,]^{\boldsymbol{T}}\,. \tag{3.53}$$

More details about these quantities will be given in Section 3.5. It is assumed that the car body roll and pitch angles are small, i.e. β, α <<1. This assumption is acceptable since the maximum value of the roll angle for passenger cars in typical rollover test maneuvers is less than 6 degrees, Forkenbrock, et al. [26], Ungoren, et al. [79, 80, 81] and Viano and Parenteau [85], in addition the pitch motion is often neglected for even roads and normal maneuvers, i.e. without sudden acceleration. Based on the theoretical derivation in Chapter 2, the computer-aided multibody system program NEWEUL can be applied resulting in the linearized equations of motion

$$\boldsymbol{M}\,\ddot{\boldsymbol{y}} + \boldsymbol{P}\,\dot{\boldsymbol{y}} + \boldsymbol{Q}\,\boldsymbol{y} = \boldsymbol{h} \tag{3.54}$$

of the spatial car model where the *mass matrix* $\boldsymbol{M}$ is given as

$$\boldsymbol{M} = \begin{bmatrix} m_S & 0 & 0 \\ 0 & \left(m_S h_{RC}{}^2 + I_X\right)\cos^2(\theta) + I_Z \sin^2(\theta) & 0 \\ 0 & 0 & I_Y \end{bmatrix}, \tag{3.55}$$

the *matrix of velocity- dependent forces* is

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\left(I_X + I_Y - I_Z\right)\dot{\gamma}\cos(\theta) \\ 0 & \left(I_X + I_Y - I_Z\right)\dot{\gamma}\cos(\theta) & 0 \end{bmatrix}, \tag{3.56}$$

the *matrix of position- dependent forces* reads as

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \begin{matrix} -m_S h_{RC} a_x \sin(\theta)\cos(\theta) \\ -\left(m_S h_{RC}^2 + I_Y - I_Z\right)\dot{\gamma}^2\cos^2(\theta) \end{matrix} & -\left(I_X - I_Z\right)\ddot{\gamma}\cos(\theta) \\ 0 & -\left(I_X - I_Z\right)\ddot{\gamma}\cos(\theta) & -\left(I_X - I_Z\right)\dot{\gamma}^2 \end{bmatrix}, \tag{3.57}$$

and *the vector of excitation forces* is represented by

$$\boldsymbol{h} = \begin{bmatrix} fz_C \\ m_S h_{RC} a_y \cos(\theta) + I_Z \ddot{\gamma}\sin(\theta) + l_\beta\left(\cos(\theta) + \alpha\sin(\theta)\right) \\ l_\alpha \end{bmatrix}. \tag{3.58}$$

The physical meaning and values of the symbols in the equations above and the NEWEUL output file are provided in Appendices A and B, respectively.

The vector of excitation forces $\boldsymbol{h}$ can be divided into two components

$$\boldsymbol{h} = \underbrace{\begin{bmatrix} 0 \\ m_S h_{RC} a_y \cos(\theta) + I_Z \ddot{\gamma} \sin(\theta) \\ 0 \end{bmatrix}}_{:= \boldsymbol{h}_w} + \underbrace{\begin{bmatrix} fz_C \\ l_\beta (\cos(\theta) + \alpha \sin(\theta)) \\ l_\alpha \end{bmatrix}}_{:= \boldsymbol{q}} \tag{3.59}$$

where the first term $\boldsymbol{h}_w$ denotes the excitation by exogenous disturbances resulting from the yaw motion of the car which generates the lateral acceleration a_y and the yaw acceleration $\ddot{\gamma}$. Summarizing these quantities in a *vector of disturbances* $\boldsymbol{w}$ yields

$$\boldsymbol{h}_w = \underbrace{\begin{bmatrix} 0 & 0 \\ m_S h_{RC} \cos(\theta) & I_Z \sin(\theta) \\ 0 & 0 \end{bmatrix}}_{:= \boldsymbol{H}_w} \underbrace{\begin{bmatrix} a_y \\ \\ \ddot{\gamma} \end{bmatrix}}_{:= \boldsymbol{w}}. \tag{3.60}$$

The second term $\boldsymbol{q}$ in Equation (3.59) is the *vector of general applied forces*, which is related to vector (3.53) by

$$\boldsymbol{q} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & (\cos(\theta) + \alpha \sin(\theta)) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{:= \boldsymbol{G}_q^\alpha} \boldsymbol{q}^a . \tag{3.61}$$

Using the equations (3.59), (3.60) and (3.61), the *linearized equations of motion of* the spatial car model (3.54) can be rewritten as

$$\boldsymbol{M}\,\ddot{\boldsymbol{y}} + \boldsymbol{P}\,\dot{\boldsymbol{y}} + \boldsymbol{Q}\,\boldsymbol{y} = \boldsymbol{H}_w \boldsymbol{w} + \boldsymbol{G}_q^\alpha\,\boldsymbol{q}^a . \tag{3.62}$$

3.5 State-space Representation of the Spatial Car Model

To analyze the control algorithms, the equations of motion are often transformed into the state-space form. In this section, the state-space representation of the spatial car model will be introduced. Based on the discussions on the control problem for the obtained plant models, the simplified spatial car model for optimal control analysis will be determined.

3.5.1 General applied forces

With small roll and pitch angle, the vector of general applied forces in the car body fixed frame $\boldsymbol{C}$ can be derived from Figure 3.8 as

$$\boldsymbol{q}^{a} := \begin{bmatrix} fz_{C} \\ l_{\beta} \\ l_{\alpha} \end{bmatrix} = \begin{bmatrix} f_{S1} + f_{S2} + f_{S3} + f_{S4} \\ t_{wf} f_{S1} - t_{wf} f_{S2} + t_{wr} f_{S3} - t_{wr} f_{S4} \\ -l_{f} f_{S1} - l_{f} f_{S2} + l_{r} f_{S3} + l_{r} f_{S4} \end{bmatrix}$$

or

$$\boldsymbol{q}^{a} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{wf} & -t_{wf} & t_{wr} & -t_{wr} \\ -l_{f} & -l_{f} & l_{r} & l_{r} \end{bmatrix}}_{:= \boldsymbol{G}} \underbrace{\begin{bmatrix} f_{S1} \\ f_{S2} \\ f_{S3} \\ f_{S4} \end{bmatrix}}_{:= \boldsymbol{f_S}} . \tag{3.63}$$

where t_{wf} and t_{wr} are the half track width of the front and rear axle, and l_f and l_r denote the distances from the car's center of gravity to the front and rear axle, respectively. The suspension forces f_{Si}, $i = 1(1)4$, sum up from spring forces F_{Ki}, damping forces F_{Bi}, anti-roll forces F_{Ai} and actuator forces (control inputs) F_{Ci}, i.e.

$$f_{Si} = F_{Ki} + F_{Bi} + F_{Ai} + F_{Ci}, \qquad i = 1(1)4,$$

which may be summarized in matrix form as

$$\underbrace{\begin{bmatrix} f_{S1} \\ f_{S2} \\ f_{S3} \\ f_{S4} \end{bmatrix}}_{\boldsymbol{f_S}} = \underbrace{\begin{bmatrix} F_{K1} \\ F_{K2} \\ F_{K3} \\ F_{K4} \end{bmatrix}}_{:= \boldsymbol{f_K}} + \underbrace{\begin{bmatrix} F_{B1} \\ F_{B2} \\ F_{B3} \\ F_{B4} \end{bmatrix}}_{:= \boldsymbol{f_B}} + \underbrace{\begin{bmatrix} F_{A1} \\ F_{A2} \\ F_{A3} \\ F_{A4} \end{bmatrix}}_{:= \boldsymbol{f_A}} + \underbrace{\begin{bmatrix} F_{C1} \\ F_{C2} \\ F_{C3} \\ F_{C4} \end{bmatrix}}_{:= \boldsymbol{u}} . \tag{3.64}$$

It should be noted that the static spring forces cancel against the weight of the vehicle and will not be considered in the following.

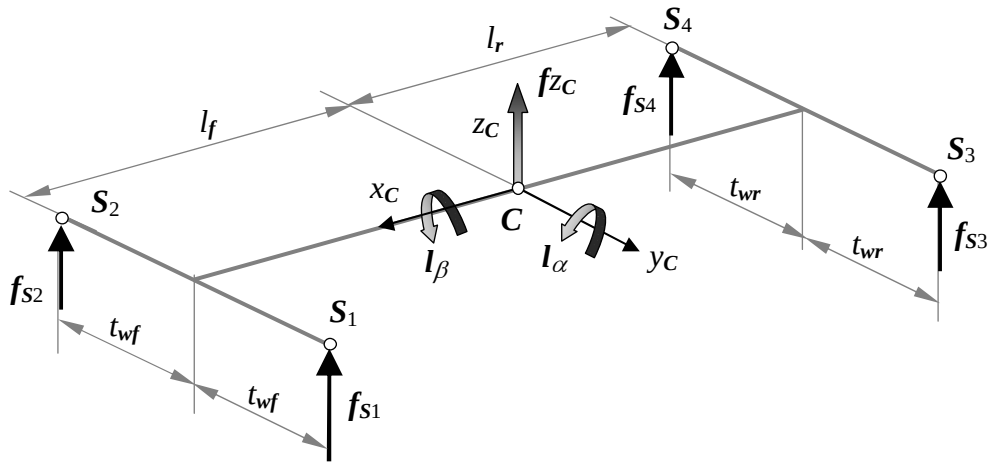

Figure 3.8: Applied forces and moments in the car body fixed frame

Since the four wheels of the car are assumed to be mass-less points keeping their traces on the road surface, i.e. $z_{Ui} = 0$, the vector of spring forces can be computed only from the vertical displacements z_{Si} of the nodal points $\boldsymbol{S_i}$ according to section 3.1.2 by

$$\boldsymbol{f_K} = -\underbrace{\begin{bmatrix} k_f & 0 & 0 & 0 \\ 0 & k_f & 0 & 0 \\ 0 & 0 & k_r & 0 \\ 0 & 0 & 0 & k_r \end{bmatrix}}_{:=\boldsymbol{K_S}} \underbrace{\begin{bmatrix} z_{S1} \\ z_{S2} \\ z_{S3} \\ z_{S4} \end{bmatrix}}_{:=\boldsymbol{z_S}}, \tag{3.65}$$

the damping forces by

$$\boldsymbol{f_B} = -\underbrace{\begin{bmatrix} b_f & 0 & 0 & 0 \\ 0 & b_f & 0 & 0 \\ 0 & 0 & b_r & 0 \\ 0 & 0 & 0 & b_r \end{bmatrix}}_{:=\boldsymbol{B_S}} \underbrace{\begin{bmatrix} \dot{z}_{S1} \\ \dot{z}_{S2} \\ \dot{z}_{S3} \\ \dot{z}_{S4} \end{bmatrix}}_{:=\dot{\boldsymbol{z}}_S} \tag{3.66}$$

and the anti-roll forces by

$$\boldsymbol{f}_A = -\underbrace{\begin{bmatrix} r_f & -r_f & 0 & 0 \\ -r_f & r_f & 0 & 0 \\ 0 & 0 & r_r & -r_r \\ 0 & 0 & -r_r & r_r \end{bmatrix}}_{:=\boldsymbol{K}_A} \underbrace{\begin{bmatrix} z_{S1} \\ z_{S2} \\ z_{S3} \\ z_{S4} \end{bmatrix}}_{:=\boldsymbol{z}_S} \tag{3.67}$$

where k_f and k_r are the translational stiffnesses of the front and rear suspensions, b_f and b_r are the damping coefficients of the front and rear suspensions, and r_f and r_r are the rotational stiffnesses of the front and rear anti-roll bars, respectively. With the above equations, the vector of total suspension forces $\boldsymbol{f}_S$ in Equation (3.64) can be expressed as a function of vertical displacements $\boldsymbol{z}_S$ and velocities $\dot{\boldsymbol{z}}_S$ of the suspension nodes attached to the sprung mass as

$$\boldsymbol{f}_S = -\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right) \boldsymbol{z}_S - \boldsymbol{B}_S \dot{\boldsymbol{z}}_S + \boldsymbol{u}. \tag{3.68}$$

For small roll and pitch angles, the linearized relations between the vertical displacements of the suspension nodes z_{Si}, i = 1(1)4, and the generalized coordinates can be defined from Figure 3.9 as

$$\begin{aligned} z_{S1} &= z_C + t_{wf}\,\beta\cos(\theta) - l_f\,\alpha, \\ z_{S2} &= z_C - t_{wf}\,\beta\cos(\theta) - l_f\,\alpha, \\ z_{S3} &= z_C + t_{wr}\,\beta\cos(\theta) + l_r\,\alpha, \\ z_{S4} &= z_C - t_{wr}\,\beta\cos(\theta) + l_r\,\alpha \end{aligned}$$

or in matrix notation

$$\underbrace{\begin{bmatrix} z_{S1} \\ z_{S2} \\ z_{S3} \\ z_{S4} \end{bmatrix}}_{\boldsymbol{z}_S} = \underbrace{\begin{bmatrix} 1 & t_{wf}\cos(\theta) & -l_f \\ 1 & -t_{wf}\cos(\theta) & -l_f \\ 1 & t_{wr}\cos(\theta) & l_r \\ 1 & -t_{wr}\cos(\theta) & l_r \end{bmatrix}}_{:=\boldsymbol{G}_S^T} \underbrace{\begin{bmatrix} z_C \\ \beta \\ \alpha \end{bmatrix}}_{\boldsymbol{y}}. \tag{3.69}$$

Due to the constant matrix $\boldsymbol{G}_S^T$, the relation between $\dot{\boldsymbol{z}}_S$ and $\dot{\boldsymbol{y}}$ reads as

$$\dot{\boldsymbol{z}}_S = \boldsymbol{G}_S^T \dot{\boldsymbol{y}}. \tag{3.70}$$

Using equations (3.63) and (3.68) − (3.70) results in the vector of general applied forces in the car body fixed frame $\boldsymbol{C}$:

$$\boldsymbol{q}^a = \boldsymbol{G}\left[-\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right)\boldsymbol{G}_S^T \boldsymbol{y} - \boldsymbol{B}_S \boldsymbol{G}_S^T \dot{\boldsymbol{y}} + \boldsymbol{u}\right]. \tag{3.71}$$

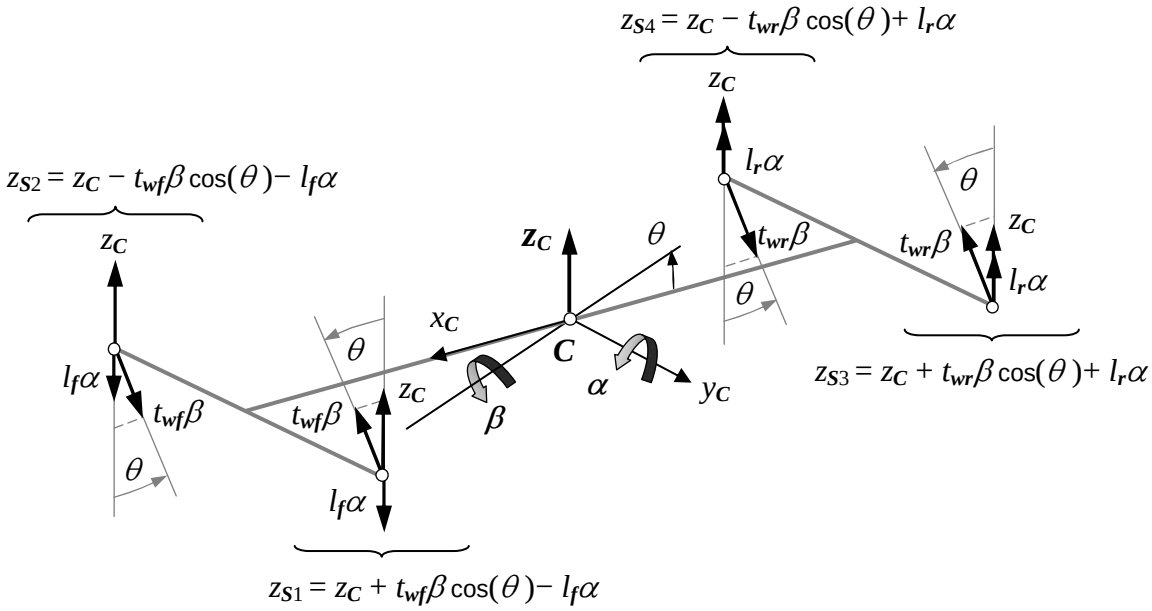

Figure 3.9: Vertical displacements of the suspension nodes for small angles α and β

Substitution of $\boldsymbol{q}^{a}$ in (3.71) with linearization due to $\alpha, \boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{u} << 1$ yields a final representation of the vector of general applied forces (3.61) in inertial system:

$$\boldsymbol{q} = \boldsymbol{G}_{\boldsymbol{q}}^{\alpha}\, \boldsymbol{G}\left[-\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right)\boldsymbol{G}_S^T\, \boldsymbol{y} - \boldsymbol{B}_S\, \boldsymbol{G}_S^T\, \dot{\boldsymbol{y}} + \boldsymbol{u}\right]$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \left(\cos(\theta) + \alpha\sin(\theta)\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{wf} & -t_{wf} & t_{wr} & -t_{wr} \\ -l_f & -l_f & l_r & l_r \end{bmatrix}$$

$$\left[-\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right)\boldsymbol{G}_S^T\, \boldsymbol{y} - \boldsymbol{B}_S\, \boldsymbol{G}_S^T\, \dot{\boldsymbol{y}} + \boldsymbol{u}\right]$$

$$= \left\{ \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{wf}\cos(\theta) & -t_{wf}\cos(\theta) & t_{wr}\cos(\theta) & -t_{wr}\cos(\theta) \\ -l_f & -l_f & l_r & l_r \end{bmatrix}}_{\boldsymbol{G}_S} + \alpha\sin(\theta) \begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{wf} & -t_{wf} & t_{wr} & -t_{wr} \\ -l_f & -l_f & l_r & l_r \end{bmatrix} \right\} \left[-\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right)\boldsymbol{G}_S^T\, \boldsymbol{y} - \boldsymbol{B}_S\, \boldsymbol{G}_S^T\, \dot{\boldsymbol{y}} + \boldsymbol{u}\right]$$

$$\approx \boldsymbol{G}_S \left[-\left(\boldsymbol{K}_S + \boldsymbol{K}_A\right)\boldsymbol{G}_S^T\, \boldsymbol{y} - \boldsymbol{B}_S\, \boldsymbol{G}_S^T\, \dot{\boldsymbol{y}} + \boldsymbol{u}\right] \qquad (3.72)$$

where $\boldsymbol{G_S}$ has been defined in Equation (3.69). The linearized equations of motion of the spatial car model (3.62) finally can be expressed by

$$\boldsymbol{M}\,\ddot{\boldsymbol{y}} + \left[\boldsymbol{P} + \boldsymbol{G_S}\boldsymbol{B_S}\boldsymbol{G_S^T}\right]\dot{\boldsymbol{y}} + \left[\boldsymbol{Q} + \boldsymbol{G_S}\left(\boldsymbol{K_S} + \boldsymbol{K_A}\right)\boldsymbol{G_S^T}\right]\boldsymbol{y} = \boldsymbol{H_w}\boldsymbol{w} + \boldsymbol{G_S}\,\boldsymbol{u}. \qquad (3.73)$$

This equation will be used in the simulation process to calculate the quantities estimating the dynamic characteristics of the spatial car model.

3.5.2 Linear parameter-varying spatial car model

In order to analyze and synthesize the control algorithms for the spatial car model, the linearized equations of motion (3.73) must be transformed to state equations as

$$\underbrace{\begin{bmatrix}\dot{\boldsymbol{y}}\\ \ddot{\boldsymbol{y}}\end{bmatrix}}_{:=\,\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{M}^{-1}\left[\boldsymbol{Q}(.) + \boldsymbol{G_S}\left(\boldsymbol{K_S} + \boldsymbol{K_A}\right)\boldsymbol{G_S^T}\right] & -\boldsymbol{M}^{-1}\left[\boldsymbol{P}(.) + \boldsymbol{G_S}\boldsymbol{B_S}\boldsymbol{G_S^T}\right]\end{bmatrix}}_{:=\,\boldsymbol{A}(a_x,\dot{\gamma},\ddot{\gamma})} \underbrace{\begin{bmatrix}\boldsymbol{y}\\ \dot{\boldsymbol{y}}\end{bmatrix}}_{:=\,\boldsymbol{x}} + \underbrace{\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{-1}\boldsymbol{G_S}\end{bmatrix}}_{:=\,\boldsymbol{B}}\boldsymbol{u} + \underbrace{\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{-1}\boldsymbol{H_w}\end{bmatrix}}_{:=\,\boldsymbol{B_w}}\underbrace{\begin{bmatrix}a_y\\ \ddot{\gamma}\end{bmatrix}}_{\boldsymbol{w}} \qquad (3.74)$$

where $\boldsymbol{x} := \left[\boldsymbol{y^T},\ \dot{\boldsymbol{y}}^{\boldsymbol{T}}\right]^{\boldsymbol{T}} = \left[z_C,\ \beta,\ \alpha,\ \dot{z}_C,\ \dot{\beta},\ \dot{\alpha}\right]^T$ is the state vector, $\boldsymbol{w}(a_y, \ddot{\gamma})$ is the vector of exogenous disturbances according to (3.60), and $\boldsymbol{A}(a_x,\dot{\gamma},\ddot{\gamma}) \in \mathbb{R}^{6\times 6}$, $\boldsymbol{B} \in \mathbb{R}^{6\times 4}$ and $\boldsymbol{B_w} \in \mathbb{R}^{6\times 2}$ are state matrices. The matrices $\boldsymbol{P}(.) := \boldsymbol{P}(\dot{\gamma})$ and $\boldsymbol{Q}(.) := \boldsymbol{Q}(a_x,\dot{\gamma},\ddot{\gamma})$ are the parameter-dependent terms of the matrix $\boldsymbol{A}(a_x,\dot{\gamma},\ddot{\gamma})$ and defined by (3.56) and (3.57), respectively.

Let $\boldsymbol{y} := \left[\boldsymbol{z_S^T},\ \ddot{\beta}\right]^{\boldsymbol{T}}$, $\boldsymbol{y} \in \mathbb{R}^5$ denote the vector of measured outputs and define the relation $\ddot{\beta} = \boldsymbol{g}_{\ddot{\beta}}^{\boldsymbol{T}}\,\ddot{\boldsymbol{y}}$ with $\boldsymbol{g}_{\ddot{\beta}}^{\boldsymbol{T}} = \left[0,\ 1,\ 0\right]$, then with (3.69) and (3.73) the measurement equations of the spatial car model can be written as

$$\underbrace{\begin{bmatrix} z_S \\ \ddot{\beta} \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} \boldsymbol{G_S^T} & \boldsymbol{0} \\ -\boldsymbol{g_{\ddot{\beta}}^T}\,\boldsymbol{M}^{-1}\left[\boldsymbol{Q}(.)+\boldsymbol{G_S}(\boldsymbol{K_S}+\boldsymbol{K_A})\boldsymbol{G_S^T}\right] & -\boldsymbol{g_{\ddot{\beta}}^T}\,\boldsymbol{M}^{-1}\left[\boldsymbol{P}(.)+\boldsymbol{G_S}\boldsymbol{B_S}\boldsymbol{G_S^T}\right] \end{bmatrix}}_{:=\boldsymbol{C}(a_x,\dot{\gamma},\ddot{\gamma})} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{\boldsymbol{x}}$$

$$+ \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g_{\ddot{\beta}}^T}\,\boldsymbol{M}^{-1}\boldsymbol{G_S} \end{bmatrix}}_{:=\boldsymbol{D}} \boldsymbol{u} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g_{\ddot{\beta}}^T}\,\boldsymbol{M}^{-1}\boldsymbol{H_w} \end{bmatrix}}_{:=\boldsymbol{D_w}} \underbrace{\begin{bmatrix} a_y \\ \ddot{\gamma} \end{bmatrix}}_{\boldsymbol{w}} \tag{3.75}$$

where $\boldsymbol{C}(a_x,\dot{\gamma},\ddot{\gamma}) \in \mathbb{R}^{5\times 6}$, $\boldsymbol{D} \in \mathbb{R}^{5\times 4}$ and $\boldsymbol{D_w} \in \mathbb{R}^{5\times 2}$ are the measurement matrices. Please note the difference in the symbols $\boldsymbol{y}$ for measurement output and $\boldsymbol{y}$ for generalized coordinate.

With the state equations (3.74) and measurement equations (3.75), the equations of motion of the spatial car model can be expressed in the state-space form of a *linear parameter-varying* (LPV) system:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(a_x,\dot{\gamma},\ddot{\gamma})\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{B_w}\,\boldsymbol{w}(a_y,\ddot{\gamma}), \\ \boldsymbol{y} = \boldsymbol{C}(a_x,\dot{\gamma},\ddot{\gamma})\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u} + \boldsymbol{D_w}\,\boldsymbol{w}(a_y,\ddot{\gamma}). \end{cases} \tag{3.76}$$

It should be pointed out that in general the varying parameters of the system can take any values in some operation region, i.e. their time functions are unknown beforehand. Therefore, the state-space matrices $\boldsymbol{A}(a_x,\dot{\gamma},\ddot{\gamma})$ and $\boldsymbol{C}(a_x,\dot{\gamma},\ddot{\gamma})$ and the vector of disturbances $\boldsymbol{w}(a_y,\ddot{\gamma})$ are treated as on-line data. The variations of the system parameters result in the changes of system dynamics. To maintain the required performance of the system, a parameter-varying controller must be designed which is able to update its parameters on-line according to the variations of the system parameters. However, for systems with more than two varying parameters, it is very difficult to design such a controller, Balas, et al. [5], Fitzpatrick [25], Gaspar, et al. [29] and Wu [93]. Therefore, the spatial car model must be simplified in order to reduce the number of system varying parameters.

In the case of given time functions for the system varying parameters a_x, a_y, $\dot{\gamma}$ and $\ddot{\gamma}$, the state-space data over the time can be computed off-line. Here the spatial car model takes on the form of a *linear time-varying* (LTV) system:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(t)\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{B}_w\,\boldsymbol{w}(t), \\ \\ \boldsymbol{y} = \boldsymbol{C}(t)\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u} + \boldsymbol{D}_w\,\boldsymbol{w}(t). \end{cases} \tag{3.77}$$

Theoretically, the optimal control problem for a linear time-varying system (3.77) can be solved with a time-varying controller as will be shown in the next chapter. This controller, however, is defined only for a specific trajectory of the system varying parameters, which can change arbitrarily in practice. Moreover, defining the time-varying controller for linear time-varying systems requires large computational effort, Ramirez [61], Schwarz [71] and Siouris [75]. For these reasons, the linear time-varying spatial car model (3.77) will not be used to design parameter-varying controller for the spatial car model.

Instead, a simplification of the spatial car model is based on constant longitudinal and angular velocities, i.e. $a_x = 0$ and $\ddot{\gamma} = 0$. With these assumptions, vector of disturbance $\boldsymbol{w}$ in equations (3.74) and (3.75) is reduced to a scalar $w = a_y$ and the *linear parameter-varying* spatial car model (3.76) becomes a *linear time-invariant* (LTI) system with constant disturbance:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(\dot{\gamma})\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{b}_w\,w(a_y), \\ \\ \boldsymbol{y} = \boldsymbol{C}(\dot{\gamma})\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u} + \boldsymbol{d}_w\,w(a_y), \qquad a_y, \dot{\gamma} = const. \end{cases} \tag{3.78}$$

Both the car yaw rate $\dot{\gamma}$ and the lateral acceleration a_y in (3.78) can be measured directly by sensors. Each pair $(\dot{\gamma}, a_y)$ defines a specific linear time-invariant system with constant disturbance, whose optimal controller can be obtained easily based on the *linear quadratic regulator* (LQR) control algorithm, which will be introduced in the next chapter in more details. By combining the optimal controllers for the linear time-invariant plants specified by selected pairs $(\dot{\gamma}, a_y)$, a parameter-dependent controller can be derived as will be shown in Chapter 6.

3.6 Simulation Model of the Spatial Car

In order to calculate the quantities estimating the dynamic characteristics of the spatial car model, a simulation model is built in MATLAB/Simulink, Moscinski and Ogonowski [52], where the spatial car model is combined with the optimal state-feedback and disturbance-feed forward controllers, Figure 3.10.

Since the varying parameters of the spatial car model can be computed from the car longitudinal velocity v and track curvature κ by equations (3.48) and (3.51), these parameters are defined as the input parameters for the simulation and generated in the block '*Maneuvers*'. By changing the time-behaviour of the car longitudinal velocity and track curvature, different operation regimes of the car are simulated. The linearized equations of the spatial car model (3.74) are solved in block '*Spatial Car Model*' by a MATLAB S-function which is presented in Appendix C. The measured outputs and dynamic criteria specified for the spatial car model are defined in block '*Criteria*'. The optimal control law with state-feedback and disturbance-feed forward is structured in block '*Controller*'. Here the control forces are computed. For the passive suspension case, the controllers are initialized with zero matrices. To observe lively the simulation process, the movement of the car is animated by block '*Animation*'.

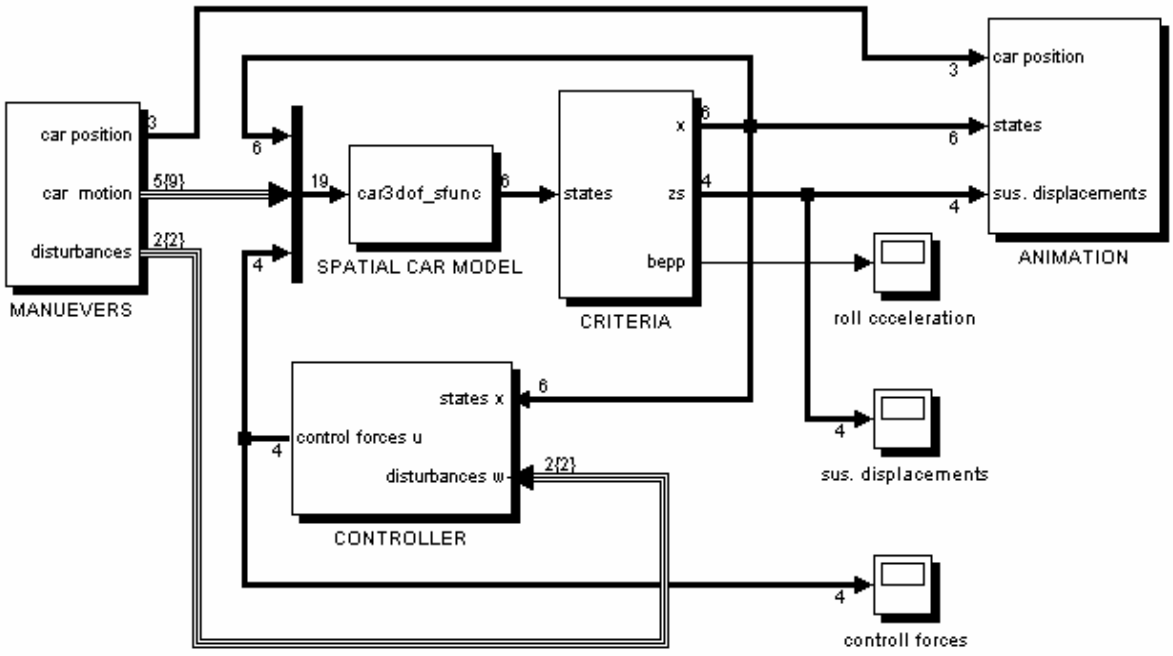

Figure 3.10: Simulation model of the spatial car model in MATLAB/Simulink

Chapter 4

Linear Quadratic Regulator Control

Linear quadratic regulator control (LQR) is a powerful concept of optimally controlling linear systems whose states are available to be fed back, i.e. the states can be measured online. In this chapter, the LQR problem is shown for *linear systems without disturbances* which results in an optimal *state-feedback controller*, and then extended to *linear systems with measurable disturbances*, which leads to an optimal *disturbance-feed forward controller*. The difficulty of solving the *differential Riccati equation* (DRE) for the case of *linear time-varying* (LTV) systems will be exposed. In contrast to the case of linear time-varying systems, the solutions of the LQR problem for *linear time-invariant* (LTI) systems can be easily obtained by solving the *algebraic Riccati equation* (ARE). The LQR control design based on the automotive performance criteria specified for the spatial car model will demonstrate the effect of LQR control to passenger cars with active suspensions.

4.1 LQR Problem for Linear Systems without Disturbances

In this section, the standard LQR problem is defined and solved based on Pontryagin's maximum principle to find the optimal controller for linear systems without disturbances. This method will be applied to linear systems with measurable disturbances in the next section to define an optimal controller for the spatial car model.

4.1.1 Definition of the LQR problem

In the case of vanishing disturbances $\boldsymbol{w}(t) = \boldsymbol{0}$, the linear time-varying spatial car model (3.77) becomes a special case of the general linear time-varying system of the form

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(t)\,\boldsymbol{x} + \boldsymbol{B}(t)\,\boldsymbol{u}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ \\ \boldsymbol{y} = \boldsymbol{C}(t)\,\boldsymbol{x} + \boldsymbol{D}(t)\,\boldsymbol{u} \end{cases} \tag{4.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{\boldsymbol{n}}$, $\boldsymbol{y}(t) \in \mathbb{R}^{\boldsymbol{m}}$, $\boldsymbol{u}(t) \in \mathbb{R}^{\boldsymbol{u}}$ are vectors of states, measured outputs and control inputs, respectively; $\boldsymbol{A}(t) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ and $\boldsymbol{B}(t) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{u}}$ are state matrices while $\boldsymbol{C}(t) \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}}$ and $\boldsymbol{D}(t) \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{u}}$ are the measurement matrices. The *LQR problem* is to find a control input $\boldsymbol{u}(t)$ minimizing the following quadratic objective function subject to constraints (4.1):

$$J = \int_0^\infty \mathcal{F}(\boldsymbol{u}, \boldsymbol{x}, t)\,dt := \int_0^\infty \left[\, \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Q}(t)\,\boldsymbol{x} + 2\,\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{N}(t)\,\boldsymbol{u} + \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{R}(t)\,\boldsymbol{u} \,\right] dt\,. \tag{4.2}$$

The matrix $\boldsymbol{Q}(t) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ is symmetric and positive semi-definite, i.e. $\boldsymbol{Q}(t) = \boldsymbol{Q}(t)^{\boldsymbol{T}} \geq 0$, weighting the states while $\boldsymbol{R}(t) \in \mathbb{R}^{\boldsymbol{u}\times\boldsymbol{u}}$ is a symmetric positive definite matrix, $\boldsymbol{R}(t) = \boldsymbol{R}(t)^{\boldsymbol{T}} > 0$, penalizing the control effort. The mix-relation between the states and control inputs denoted by the matrix $\boldsymbol{N}(t) \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{u}}$ will be shown later in Equation (4.4).

Relatively small elements of $\boldsymbol{Q}(t)$ compared to $\boldsymbol{R}(t)$ will result in a control law which will tolerate large errors in the states with low inputs. On the other hand, if $\boldsymbol{Q}(t)$ is made large compared to $\boldsymbol{R}(t)$, this will result in tight control, i.e. small errors in the states with considerable inputs. Different values of the entries of $\boldsymbol{Q}(t)$ or $\boldsymbol{R}(t)$ can be used to penalize specific states compared to others at the same expense of control energy. In the standard LQR problem the matrix $\boldsymbol{R}(t)$ is assumed to be positive definite, however, the special case $\boldsymbol{R}(t) = \boldsymbol{0}$ of the LQR problem called the *singular problem* can be found in Moylan and Moore [53], Ramirez [61] and Schwarz [71].

In most optimization problems, the objectives that need to be regarded for minimization are not the complete state vector $\boldsymbol{x}(t)$ but only a selection according to measurement outputs $\boldsymbol{y}(t)$. Therefore, the objective function (4.2) may be replaced by

$$J = \int_0^\infty \left[\, \boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{Q}_{\boldsymbol{y}}(t)\,\boldsymbol{y} + \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{R}_{\boldsymbol{u}}(t)\,\boldsymbol{u} \,\right] dt\,. \tag{4.3}$$

where $\boldsymbol{Q}_y(t) \in \mathbb{R}^{m \times m}$, $\boldsymbol{Q}_y(t) = \boldsymbol{Q}_y(t)^T \geq 0$ and $\boldsymbol{R}_u(t) \in \mathbb{R}^{u \times u}$, $\boldsymbol{R}_u(t) = \boldsymbol{R}_u(t)^T > 0$ are weighting matrices of the measured outputs and controlled inputs, respectively.

Substitution of $\boldsymbol{y}(t)$ from (4.1) in (4.3) yields

$$\begin{aligned} J &= \int_0^\infty \left[(\boldsymbol{C}\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u})^T \boldsymbol{Q}_y (\boldsymbol{C}\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u}) + \boldsymbol{u}^T \boldsymbol{R}_u \boldsymbol{u} \right] dt \\ &= \int_0^\infty \left[\boldsymbol{x}^T \underbrace{\left(\boldsymbol{C}^T \boldsymbol{Q}_y \boldsymbol{C}\right)}_{:=\boldsymbol{Q}(t)} \boldsymbol{x} + 2\boldsymbol{x}^T \underbrace{\left(\boldsymbol{C}^T \boldsymbol{Q}_y \boldsymbol{D}\right)}_{:=\boldsymbol{N}(t)} \boldsymbol{u} + \boldsymbol{u}^T \underbrace{\left(\boldsymbol{D}^T \boldsymbol{Q}_y \boldsymbol{D} + \boldsymbol{R}_u\right)}_{:=\boldsymbol{R}(t)} \boldsymbol{u} \right] dt \\ &= \int_0^\infty \left[\boldsymbol{x}^T \boldsymbol{Q}\,\boldsymbol{x} + 2\,\boldsymbol{x}^T \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{R}\,\boldsymbol{u} \right] dt \,. \end{aligned} \tag{4.4}$$

The equation shows that the objective function J defined in (4.3) is contained in the more general form (4.2) with properly defined matrices $\boldsymbol{Q}(t)$, $\boldsymbol{N}(t)$, and $\boldsymbol{R}(t)$. The assumptions of $\boldsymbol{Q}_y(t) \geq 0$ and $\boldsymbol{R}_u(t) > 0$ ensure the conditions of $\boldsymbol{Q}(t) \geq 0$ and $\boldsymbol{R}(t) > 0$ to hold for any $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$.

4.1.2 LQR solution using the Pontryagin's maximum principle

There are different techniques to solve the LQR problem. In this section, the solution of the LQR problem will be deduced from Pontryagin's maximum principle. *Pontryagin's maximum principle* states that the optimal control $\boldsymbol{u}(t)$ that minimizes the objective function (4.2) subject to constraint (4.1) must minimize the so-called *Hamiltonian*

$$\begin{aligned} \mathcal{H}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}, t) &:= \mathcal{F}(\boldsymbol{u}, \boldsymbol{x}, t) + \boldsymbol{\lambda}^T (\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u}) \\ &= \left(\boldsymbol{x}^T \boldsymbol{Q}\,\boldsymbol{x} + 2\,\boldsymbol{x}^T \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{R}\,\boldsymbol{u}\right) + \boldsymbol{\lambda}^T (\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u}) \end{aligned} \tag{4.5}$$

where $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$ are the *dynamic Lagrange multipliers* or *co-state vector*. According to (4.2) and (4.5) the objective function can also be described as

$$\begin{aligned} J(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}, t) &= \int_0^\infty \mathcal{F}(\boldsymbol{u}, \boldsymbol{x}, t)\, dt = \int_0^\infty \left[\mathcal{H} - \boldsymbol{\lambda}^T \underbrace{(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u})}_{\dot{\boldsymbol{x}}} \right] dt \\ &= -\int_0^\infty \boldsymbol{\lambda}^T \dot{\boldsymbol{x}}\, dt + \int_0^\infty \mathcal{H}\, dt \end{aligned}$$

or after partial integration of the first term

$$J(\boldsymbol{u},\boldsymbol{x},\boldsymbol{\lambda},t) = -\boldsymbol{\lambda}^{\boldsymbol{T}}\boldsymbol{x}\Big|_0^{\infty} + \int_0^{\infty} \dot{\boldsymbol{\lambda}}^{\boldsymbol{T}}\boldsymbol{x}\,dt + \int_0^{\infty} \mathcal{H}\,dt$$

$$= \boldsymbol{\lambda}_0^{\boldsymbol{T}}\boldsymbol{x}_0 + \int_0^{\infty} \dot{\boldsymbol{\lambda}}^{\boldsymbol{T}}\boldsymbol{x}\;dt + \int_0^{\infty} \mathcal{H}\,dt \qquad (4.6)$$

where the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and the asymptotic value $\boldsymbol{x}(\infty) = \boldsymbol{0}$ for asymptotic stable systems are used. By introducing the variations $\delta\boldsymbol{x}$ and $\delta\boldsymbol{u}$ of state and control input, we can find the expansion of the objective function in a first order Taylor-series about the optimal point:

$$J(\boldsymbol{u}+\delta\boldsymbol{u},\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{\lambda},t) \approx J(\boldsymbol{u},\boldsymbol{x},\boldsymbol{\lambda},t) + \left[\left(\frac{\partial J}{\partial \boldsymbol{x}}\right)^{\boldsymbol{T}}\delta\boldsymbol{x} + \left(\frac{\partial J}{\partial \boldsymbol{u}}\right)^{\boldsymbol{T}}\delta\boldsymbol{u}\right]. \qquad (4.7)$$

The *necessary condition* for a *local extremum* of J is that the first order term of the Taylor-series must vanish:

$$\delta J = J(\boldsymbol{u}+\delta\boldsymbol{u},\boldsymbol{x}+\delta\boldsymbol{x},\boldsymbol{\lambda},t) - J(\boldsymbol{u},\boldsymbol{x},\boldsymbol{\lambda},t)$$

$$= \left[\left(\frac{\partial J}{\partial \boldsymbol{x}}\right)^{\boldsymbol{T}}\delta\boldsymbol{x} + \left(\frac{\partial J}{\partial \boldsymbol{u}}\right)^{\boldsymbol{T}}\delta\boldsymbol{u}\right] \overset{!}{=} 0. \qquad (4.8)$$

By the introduction of the co-state vector $\boldsymbol{\lambda}(t)$ this condition is satisfied for arbitrary variations $\delta\boldsymbol{x}$ and $\delta\boldsymbol{u}$ only if

$$\frac{\partial J}{\partial \boldsymbol{x}} \overset{!}{=} 0 \qquad (4.9)$$

and

$$\frac{\partial J}{\partial \boldsymbol{u}} \overset{!}{=} 0. \qquad (4.10)$$

With (4.6) equation (4.9) leads to the so-called *Euler-Lagrange equation*

$$\frac{\partial J}{\partial \boldsymbol{x}} \overset{!}{=} \boldsymbol{0} = \int_0^{\infty} \dot{\boldsymbol{\lambda}}\,dt + \int_0^{\infty} \frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}\,dt \quad \Rightarrow \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}, \qquad (4.11)$$

or with (4.5):

$$\dot{\boldsymbol{\lambda}} = -2\,\boldsymbol{Q}\,\boldsymbol{x} - 2\,\boldsymbol{N}\,\boldsymbol{u} - \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{\lambda}. \qquad (4.12)$$

From (4.10) and (4.5) we find

$$\frac{\partial J}{\partial \boldsymbol{u}} \overset{!}{=} \boldsymbol{0} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} \tag{4.13}$$

$$= 2\,\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{x} + 2\,\boldsymbol{R}\,\boldsymbol{u} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{\lambda}, \tag{4.14}$$

which can be solved for the optimal control as

$$\boldsymbol{u}^{*} = -\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{\lambda}\right). \tag{4.15}$$

Note that $\boldsymbol{R}(t)^{-1}$ is assured to exist due to $\boldsymbol{R}(t) > 0$. This is known as the *sufficient condition* for a local minimum, which can be obtained by letting $\partial^2 J / \partial^2 \boldsymbol{u} > 0$. Equation (4.15) shows that the optimal control $\boldsymbol{u}(t)$ is a function of the state vector $\boldsymbol{x}(t)$ and the co-state vector $\boldsymbol{\lambda}(t)$. In order to eliminate $\boldsymbol{\lambda}(t)$ in (4.15), we assume the *Riccati transformation*

$$\boldsymbol{\lambda} = 2\,\boldsymbol{P}\,\boldsymbol{x} \tag{4.16}$$

where $\boldsymbol{P}(t)$ is called the *Riccati matrix*. Using the Riccati transformation, the optimal control becomes

$$\boldsymbol{u}^{*} = -\underbrace{\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right)}_{:=\,\boldsymbol{K}^{*}(t)}\boldsymbol{x} = -\boldsymbol{K}^{*}(t)\,\boldsymbol{x} \tag{4.17}$$

demonstrating that the optimal control is a time-varying, proportional *state feedback controller* with gains $\boldsymbol{K}^{*}(t) \in \mathbb{R}^{\boldsymbol{u}\times\boldsymbol{n}}$. The optimal control structure for the standard LQR problem is illustrated in Figure 4.1. To compute $\boldsymbol{K}^{*}(t)$, the Riccati matrix $\boldsymbol{P}(t)$ must be determined.

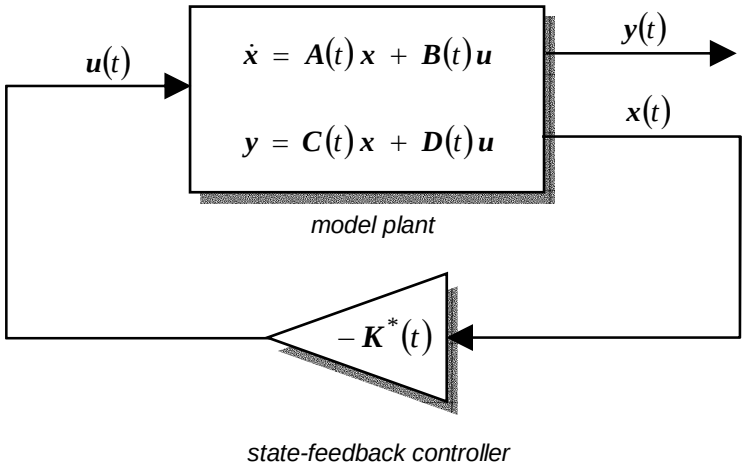

Figure 4.1: Optimal state-feedback control for linear systems without disturbances

4.1.3 Algebraic Riccati equation

Taking the derivative of the Riccati transformation (4.16) gives

$$\dot{\boldsymbol{\lambda}} = 2\,\boldsymbol{P}\,\dot{\boldsymbol{x}} + 2\,\dot{\boldsymbol{P}}\,\boldsymbol{x}\,. \tag{4.18}$$

Substitution of the Riccati transformation (4.16) in the Euler-Lagrange equation (4.12) gives

$$\dot{\boldsymbol{\lambda}} = -2\,\boldsymbol{Q}\,\boldsymbol{x} - 2\,\boldsymbol{N}\,\boldsymbol{u} - 2\,\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P}\,\boldsymbol{x}\,. \tag{4.19}$$

Equating these two expressions for $\dot{\boldsymbol{\lambda}}(t)$ yields

$$\boldsymbol{P}\,\dot{\boldsymbol{x}} + \dot{\boldsymbol{P}}\,\boldsymbol{x} = -\boldsymbol{Q}\,\boldsymbol{x} - \boldsymbol{N}\,\boldsymbol{u} - \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P}\,\boldsymbol{x}\,. \tag{4.20}$$

With the state equations (4.1) and the control law (4.17) we find

$$\begin{aligned} -\dot{\boldsymbol{P}}\,\boldsymbol{x} &= \boldsymbol{P}\left(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u}\right) + \boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P}\,\boldsymbol{x} \\ &= \boldsymbol{P}\left\{\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right)\right]\boldsymbol{x}\right\} + \boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{N}\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right)\right]\boldsymbol{x} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P}\,\boldsymbol{x}\,. \end{aligned} \tag{4.21}$$

This is valid for any state vector $\boldsymbol{x}(t)$ only if the following matrix equality holds:

$$-\dot{\boldsymbol{P}} = \boldsymbol{P}\,\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P} - \left(\boldsymbol{P}\,\boldsymbol{B} + \boldsymbol{N}\right)\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right) + \boldsymbol{Q}\,. \tag{4.22}$$

Equation (4.22) is called the *differential Riccati equation* (DRE) and is a nonlinear first order differential equation with a time-dependent solution $\boldsymbol{P}(t)$, Ramirez [61], Schwarz [71] and Siouris [75]. For the *infinite-time* LQR problem Schwarz [71] proved that $\boldsymbol{P}(t)$ approaches a constant matrix $\boldsymbol{P}$, i.e.

$$\lim_{t\to\infty} \boldsymbol{P}(t) = \boldsymbol{P},\ \ \boldsymbol{P} = \boldsymbol{P}^{\boldsymbol{T}} \geq 0\,. \tag{4.23}$$

Hence $\dot{\boldsymbol{P}}(t) = \boldsymbol{0}$ and Equation (4.22) becomes a time-dependent algebraic Riccati equation. Since this equation does not depend on the state $\boldsymbol{x}(t)$ or control $\boldsymbol{u}(t)$, it can be computed independently from the state differential equations which means that the optimal control gain matrix $\boldsymbol{K}^{*}(t)$ can be computed separately from the state dynamic response. In practice, it will be pre-computed and stored for later use.

For *linear time-invariant* (LTI) systems where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{Q}$, $\boldsymbol{N}$ and $\boldsymbol{R}$ are constant matrices, the control task can be simplified. In this case, the unique solution $\boldsymbol{P}$ of the time-invariant *algebraic Riccati equation* (ARE)

$$\boldsymbol{PA} + \boldsymbol{A}^T\boldsymbol{P} - (\boldsymbol{PB} + \boldsymbol{N})\boldsymbol{R}^{-1}(\boldsymbol{N}^T + \boldsymbol{B}^T\boldsymbol{P}) + \boldsymbol{Q} = \boldsymbol{0} \tag{4.24}$$

results in the optimal *state-feedback controller* with a constant *state-feedback gain matrix* $\boldsymbol{K}^*$ defined by (4.17):

$$\boldsymbol{K}^* = \boldsymbol{R}^{-1}(\boldsymbol{N}^T + \boldsymbol{B}^T\boldsymbol{P}). \tag{4.25}$$

The conditions for existence and uniqueness of the stabilizing optimal control known as the *Kalman criteria* for controllability and observability, Ackermann [1], Ramirez [61] and Schwarz [71], are as follows:

- the pair ($\boldsymbol{A}$, $\boldsymbol{B}$) is controllable, i.e.

$$Rank\left[\boldsymbol{B},\ \boldsymbol{AB},\ ...,\ \boldsymbol{A}^{n-1}\boldsymbol{B}\right] = n, \tag{4.26}$$

- the pair $\left(\boldsymbol{Q}_S := \boldsymbol{Q} - \boldsymbol{N}\boldsymbol{R}^{-1}\boldsymbol{N}^T,\ \boldsymbol{A}_S := \boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{N}^T\right)$ is observable, i.e.

$$Rank\left[\boldsymbol{Q}_S^T,\ \boldsymbol{A}_S^T\boldsymbol{Q}_S^T,\ ...,\ \left(\boldsymbol{A}_S^T\right)^{n-1}\boldsymbol{Q}_S^T\right] = n, \tag{4.27}$$

Within the MATLAB software, Shahian and Hassul [72], the LQR problem for linear time-invariant systems can be solved for the objective functions defined by (4.2) and (4.3) using the command '*lqr*' and '*lqry*', respectively. Once the weighting matrices $\boldsymbol{Q}$, $\boldsymbol{N}$ and $\boldsymbol{R}$ in (4.2) or $\boldsymbol{Q}_y$ and $\boldsymbol{R}_u$ in (4.3) are defined, the commands '*lqr*' and '*lqry*' will check the conditions (4.26) and (4.27) automatically and return the Riccati matrix $\boldsymbol{P}$ and the optimal state-feedback gain matrix $\boldsymbol{K}$ if those conditions are satisfied. The optimal control $\boldsymbol{u}$ can then be derived easily by (4.17). The weighting matrices, therefore, can be considered as the design parameters of the LQR problem.

4.2 LQR for Linear Systems with Measurable Disturbances

The standard LQR approach is used to synthesise an optimal controller for systems without disturbances. In practice, however, most systems are influenced by exogenous disturbances which also effect control optimality. In this section, the LQR problem for linear systems excited by measurable disturbances will be presented.

4.2.1 Problem definition

Let us consider the linear time-varying system (3.77) including disturbances in a more general form, i.e.

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u} + \boldsymbol{B}_w(t)\boldsymbol{w}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ \boldsymbol{y} = \boldsymbol{C}(t)\boldsymbol{x} + \boldsymbol{D}(t)\boldsymbol{u} + \boldsymbol{D}_w(t)\boldsymbol{w} \end{cases} \tag{4.28}$$

where $\boldsymbol{B}_w(t) \in \mathbb{R}^{n \times w}$ and $\boldsymbol{D}_w(t) \in \mathbb{R}^{m \times w}$. The vector of disturbances $\boldsymbol{w}(t) \in \mathbb{R}^w$ is assumed to be measurable in real time. The LQR problem for the system (4.28) is to find a control input $\boldsymbol{u}(t)$ minimizing the objective function J defined by (4.3).

Substituting $\boldsymbol{y}(t)$ from (4.28) into (4.3) yields

$$\begin{aligned} J &= \int_0^\infty \left[(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{D}_w\boldsymbol{w})^T \boldsymbol{Q}_y (\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{D}_w\boldsymbol{w}) + \boldsymbol{u}^T \boldsymbol{R}_u \boldsymbol{u} \right] dt \\ &= \int_0^\infty \left[(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u})^T \boldsymbol{Q}_y (\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}) + 2(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u})^T \boldsymbol{Q}_y \boldsymbol{D}_w \boldsymbol{w} + \boldsymbol{w}^T \boldsymbol{D}_w^T \boldsymbol{Q}_y \boldsymbol{D}_w \boldsymbol{w} + \boldsymbol{u}^T \boldsymbol{R}_u \boldsymbol{u} \right] dt \\ &= \int_0^\infty \Big[\boldsymbol{x}^T \underbrace{\left(\boldsymbol{C}^T \boldsymbol{Q}_y \boldsymbol{C}\right)}_{:=\boldsymbol{Q}(t)} \boldsymbol{x} + 2\boldsymbol{x}^T \underbrace{\left(\boldsymbol{C}^T \boldsymbol{Q}_y \boldsymbol{D}\right)}_{:=\boldsymbol{N}(t)} \boldsymbol{u} + \boldsymbol{u}^T \underbrace{\left(\boldsymbol{D}^T \boldsymbol{Q}_y \boldsymbol{D} + \boldsymbol{R}_u\right)}_{:=\boldsymbol{R}(t)} \boldsymbol{u} \Big] dt \\ &\quad + \int_0^\infty \Big\{ 2\boldsymbol{x}^T \underbrace{\boldsymbol{C}^T \boldsymbol{Q}_y \boldsymbol{D}_w}_{:=\boldsymbol{N}_{xw}(t)} \boldsymbol{w} + 2\boldsymbol{w}^T \underbrace{\boldsymbol{D}_w^T \boldsymbol{Q}_y \boldsymbol{D}}_{:=\boldsymbol{N}_{uw}(t)} \boldsymbol{u} + \boldsymbol{w}^T \underbrace{\boldsymbol{D}_w^T \boldsymbol{Q}_y \boldsymbol{D}_w}_{:=\boldsymbol{R}_w(t)} \boldsymbol{w} \Big\} dt \\ &= \int_0^\infty \left[\left(\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{x}^T \boldsymbol{N}\boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{R}\boldsymbol{u}\right) + \left(2\boldsymbol{x}^T \boldsymbol{N}_{xw}\boldsymbol{w} + 2\boldsymbol{w}^T \boldsymbol{N}_{uw}\boldsymbol{u} + \boldsymbol{w}^T \boldsymbol{R}_w \boldsymbol{w}\right) \right] dt \\ &=: \int_0^\infty \mathcal{F}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{w}, t)\, dt. \end{aligned} \tag{4.29}$$

Different from the standard LQR problem is the second group of terms depending on the disturbances $\boldsymbol{w}(t)$. Therefore, the control law has to be a function of both $\boldsymbol{x}(t)$ and $\boldsymbol{w}(t)$. It should be noted that the solution of the optimal control problem for disturbed linear systems via the LQR approach requires both the state $\boldsymbol{x}(t)$ and disturbance $\boldsymbol{w}(t)$ to be known or measurable over the entire time domain.

4.2.2 Solution based on the Pontryagin's maximum principle

The *Hamiltonian* for the optimal control problem of linear systems (4.28) with measurable disturbances can be defined as

$$\begin{aligned}\mathcal{H}(\boldsymbol{u},\boldsymbol{x},\boldsymbol{w},\boldsymbol{\lambda},t) &:= \mathcal{F}(\boldsymbol{u},\boldsymbol{x},\boldsymbol{w},t) + \boldsymbol{\lambda}^{\boldsymbol{T}}\left(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{B}_{\boldsymbol{w}}\,\boldsymbol{w}\right) \\ &= \left(\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Q}\,\boldsymbol{x} + 2\,\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{R}\,\boldsymbol{u}\right) + \left(2\,\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{xw}}\boldsymbol{w} + 2\,\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{uw}}\boldsymbol{u} + \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{w}\right) \\ &\quad + \boldsymbol{\lambda}^{\boldsymbol{T}}\left(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{B}_{\boldsymbol{w}}\,\boldsymbol{w}\right). \end{aligned} \tag{4.30}$$

Based on the Pontryagin's maximum principle, the *necessary condition* for a *local minimum* is given by the *Euler-Lagrange equation*

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}} = -\left(2\,\boldsymbol{Q}\boldsymbol{x} + 2\,\boldsymbol{N}\boldsymbol{u} + 2\,\boldsymbol{N}_{\boldsymbol{xw}}\boldsymbol{w} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{\lambda}\right) \tag{4.31}$$

and the *optimal control equation*

$$\boldsymbol{0} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = 2\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{x} + 2\boldsymbol{R}\boldsymbol{u} + 2\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}}\boldsymbol{w} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{\lambda}. \tag{4.32}$$

The optimal control can be derived from (4.32) as

$$\boldsymbol{u}^{*} = -\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}}\boldsymbol{w} + \frac{1}{2}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{\lambda}\right). \tag{4.33}$$

In order to eliminate the co-state vector $\boldsymbol{\lambda}(t)$ from Equation (4.33), the *Riccati transformation* is assumed as

$$\boldsymbol{\lambda} = 2\left(\boldsymbol{P}\,\boldsymbol{x} - \boldsymbol{\xi}\right) \tag{4.34}$$

where $\boldsymbol{P}(t) = \boldsymbol{P}^{\boldsymbol{T}}(t)$ and $\boldsymbol{\xi}(t)$ are the so-called *Riccati matrix* and *Riccati vector*, respectively. Then the optimal control becomes

$$\boldsymbol{u}^{*} = -\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{T}\boldsymbol{x} + \boldsymbol{N}_{uw}^{T}\boldsymbol{w} + \boldsymbol{B}^{T}\boldsymbol{P}\,\boldsymbol{x} - \boldsymbol{B}^{T}\boldsymbol{\xi}\right)$$

$$= \underbrace{\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{T} + \boldsymbol{B}^{T}\boldsymbol{P}\right)\boldsymbol{x}\right]}_{:=\boldsymbol{u}_{x}^{*}(t)} + \underbrace{\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}_{uw}^{T}\boldsymbol{w} - \boldsymbol{B}^{T}\boldsymbol{\xi}\right)\right]}_{:=\boldsymbol{u}_{w}^{*}(t)}. \qquad (4.35)$$

This equation shows that the optimal control $\boldsymbol{u}(t)^{*}$ for the stated problem includes not only *state-feedback control* $\boldsymbol{u}_{x}^{*}(t)$, but also *disturbance feed-forward control* $\boldsymbol{u}_{w}^{*}(t)$. To complete the optimal control $\boldsymbol{u}(t)^{*}$, the Riccati matrix $\boldsymbol{P}(t)$ and Riccati vector $\boldsymbol{\xi}(t)$ must be defined.

Taking the derivative of the Riccati transformation (4.34) gives

$$\dot{\boldsymbol{\lambda}} = 2\left(\boldsymbol{P}\,\dot{\boldsymbol{x}} + \dot{\boldsymbol{P}}\,\boldsymbol{x} - \dot{\boldsymbol{\xi}}\right). \qquad (4.36)$$

Substituting the Riccati transformation (4.34) in the Euler-Lagrange equation (4.31) yields

$$\dot{\boldsymbol{\lambda}} = -2\left[\boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{N}_{xw}\,\boldsymbol{w} + \boldsymbol{A}^{T}\left(\boldsymbol{P}\,\boldsymbol{x} - \boldsymbol{\xi}\right)\right]. \qquad (4.37)$$

Equating these two expressions for $\dot{\boldsymbol{\lambda}}(t)$ yields

$$\dot{\boldsymbol{P}}\,\boldsymbol{x} + \boldsymbol{P}\,\dot{\boldsymbol{x}} - \dot{\boldsymbol{\xi}} = -\left[\boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{N}_{xw}\boldsymbol{w} + \boldsymbol{A}^{T}\left(\boldsymbol{P}\,\boldsymbol{x} - \boldsymbol{\xi}\right)\right]. \qquad (4.38)$$

The state equations (4.28) yield

$$\dot{\boldsymbol{P}}\,\boldsymbol{x} + \boldsymbol{P}\left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{B}_{w}\boldsymbol{w}\right) - \dot{\boldsymbol{\xi}} = -\left[\boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{N}\,\boldsymbol{u} + \boldsymbol{N}_{xw}\boldsymbol{w} + \boldsymbol{A}^{T}\left(\boldsymbol{P}\,\boldsymbol{x} - \boldsymbol{\xi}\right)\right]$$

or after rearranging

$$\left(\dot{\boldsymbol{P}}\,\boldsymbol{x} + \boldsymbol{P}\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{A}^{T}\boldsymbol{P}\,\boldsymbol{x} + \boldsymbol{Q}\,\boldsymbol{x}\right) + \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{u} - \left[\dot{\boldsymbol{\xi}} + \boldsymbol{A}^{T}\boldsymbol{\xi} - \left(\boldsymbol{N}_{xw} + \boldsymbol{P}\boldsymbol{B}_{w}\right)\boldsymbol{w}\right] = \boldsymbol{0}.$$

With the control law (4.35), the equation becomes

$$\left(\dot{\boldsymbol{P}}\,\boldsymbol{x} + \boldsymbol{P}\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{A}^{T}\boldsymbol{P}\,\boldsymbol{x} + \boldsymbol{Q}\,\boldsymbol{x}\right) + \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{T} + \boldsymbol{B}^{T}\boldsymbol{P}\right)\boldsymbol{x}\right]$$
$$+ \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\left[-\boldsymbol{R}^{-1}\left(\boldsymbol{N}_{uw}^{T}\boldsymbol{w} - \boldsymbol{B}^{T}\boldsymbol{\xi}\right)\right] - \left[\dot{\boldsymbol{\xi}} + \boldsymbol{A}^{T}\boldsymbol{\xi} - \left(\boldsymbol{N}_{xw} + \boldsymbol{P}\boldsymbol{B}_{w}\right)\boldsymbol{w}\right] = \boldsymbol{0}$$

or

$$\left[\dot{\boldsymbol{P}} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P} - \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right) + \boldsymbol{Q}\right]\boldsymbol{x}$$
$$-\dot{\boldsymbol{\xi}} - \left[\boldsymbol{A}^{\boldsymbol{T}} + \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\right]\boldsymbol{\xi} - \left[\left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{R}^{-1}\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} - \left(\boldsymbol{N}_{\boldsymbol{xw}} + \boldsymbol{P}\boldsymbol{B}_{\boldsymbol{w}}\right)\right]\boldsymbol{w} = \boldsymbol{0}\,. \tag{4.39}$$

Equation (4.39) is satisfied for any vector of states $\boldsymbol{x}(t)$ and disturbances $\boldsymbol{w}(t)$ only if the two lines vanish separately. This results in the already known *differential Riccati matrix equation* (4.22), i.e.

$$\dot{\boldsymbol{P}} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P} - \left(\boldsymbol{P}\boldsymbol{B} + \boldsymbol{N}\right)\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right) + \boldsymbol{Q} = \boldsymbol{0}\,, \tag{4.40}$$

and the so-called *differential Riccati vector equation*

$$\dot{\boldsymbol{\xi}} + \left[\boldsymbol{A}^{\boldsymbol{T}} + \left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\right]\boldsymbol{\xi} + \left[\left(\boldsymbol{N} + \boldsymbol{P}\boldsymbol{B}\right)\boldsymbol{R}^{-1}\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} - \left(\boldsymbol{N}_{\boldsymbol{xw}} + \boldsymbol{P}\boldsymbol{B}_{\boldsymbol{w}}\right)\right]\boldsymbol{w} = \boldsymbol{0}\,. \tag{4.41}$$

Let $\boldsymbol{K}_{\boldsymbol{x}}^{*}(t)$ denote the optimal *state-feedback controller* where according to Equation (4.35)

$$\boldsymbol{K}_{\boldsymbol{x}}^{*} = -\boldsymbol{R}^{-1}\left(\boldsymbol{N}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\right), \quad \boldsymbol{K}_{\boldsymbol{x}}^{*}(t) \in \mathbb{R}^{\boldsymbol{u}\times\boldsymbol{n}}\,, \tag{4.42}$$

the differential Riccati vector equation (4.41) can be simplified to

$$\dot{\boldsymbol{\xi}} + \left[\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{K}_{\boldsymbol{x}}^{*\,\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\right]\boldsymbol{\xi} + \left[\boldsymbol{K}_{\boldsymbol{x}}^{*\,\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} - \left(\boldsymbol{N}_{\boldsymbol{xw}} + \boldsymbol{P}\boldsymbol{B}_{\boldsymbol{w}}\right)\right]\boldsymbol{w} = \boldsymbol{0}\,. \tag{4.43}$$

For the *infinite-time* problem we get $\dot{\boldsymbol{P}}(t) = \boldsymbol{0}$ and $\dot{\boldsymbol{\xi}}(t) = \boldsymbol{0}$, see Ramirez [61] and Schwarz [71]. Equations (4.40) and (4.43) then become the time-varying algebraic equations resulting in the time-dependent Riccati matrix $\boldsymbol{P}(t)$ and vector $\boldsymbol{\xi}(t)$, respectively. In this case the Riccati vector $\boldsymbol{\xi}(t)$ can be computed by

$$\boldsymbol{\xi} = -\left[\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{K}_{\boldsymbol{x}}^{*\,\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\right]^{-1}\left[\boldsymbol{K}_{\boldsymbol{x}}^{*\,\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} - \left(\boldsymbol{N}_{\boldsymbol{xw}} + \boldsymbol{P}\boldsymbol{B}_{\boldsymbol{w}}\right)\right]\boldsymbol{w}\,. \tag{4.44}$$

With the state-feedback controller $\boldsymbol{K}_{\boldsymbol{x}}^{*}(t)$ and the Riccati vector $\boldsymbol{\xi}(t)$ defined by (4.42) and (4.44), respectively, the disturbance feed-forward control $\boldsymbol{u}_{\boldsymbol{w}}^{*}(t)$ in (4.35) can be written as

$$\boldsymbol{u}_{\boldsymbol{w}}^{*} = \underbrace{-\boldsymbol{R}^{-1}\left\{\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} + \boldsymbol{B}^{\boldsymbol{T}}\left[\boldsymbol{A}^{\boldsymbol{T}} + \left(\boldsymbol{K}_{\boldsymbol{x}}^{*}\right)^{\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\right]^{-1}\left[\left(\boldsymbol{K}_{\boldsymbol{x}}^{*}\right)^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{uw}}^{\boldsymbol{T}} - \left(\boldsymbol{N}_{\boldsymbol{xw}} + \boldsymbol{P}\boldsymbol{B}_{\boldsymbol{w}}\right)\right]\right\}}_{:=\,\boldsymbol{K}_{\boldsymbol{w}}^{*}(t)\in\,\mathbb{R}^{\boldsymbol{u}\times\boldsymbol{w}}}\boldsymbol{w} \tag{4.45}$$

where $\boldsymbol{K}_w^*(t)$ represents the optimal *disturbance-feed forward controller*. Consequently, the optimal control is defined by

$$\boldsymbol{u}^* = \boldsymbol{u}_x^* + \boldsymbol{u}_w^* = -\boldsymbol{K}_x^* \boldsymbol{x} - \boldsymbol{K}_w^* \boldsymbol{w}. \tag{4.46}$$

For the case of *linear time-invariant* (LTI) systems, $\boldsymbol{P}(t) = \boldsymbol{P}$ is a constant matrix which is the unique solution of the algebraic Riccati equation (4.24) resulting in the constant optimal gain matrices $\boldsymbol{K}_x^*$ and $\boldsymbol{K}_w^*$.

Figure 4.2 shows the optimal control structure for linear systems with measurable disturbances. It can be realized clearly that the standard LQR problem described in Section 4.1 is a special case of the stated problem with $\boldsymbol{w}(t) = \boldsymbol{0}$.

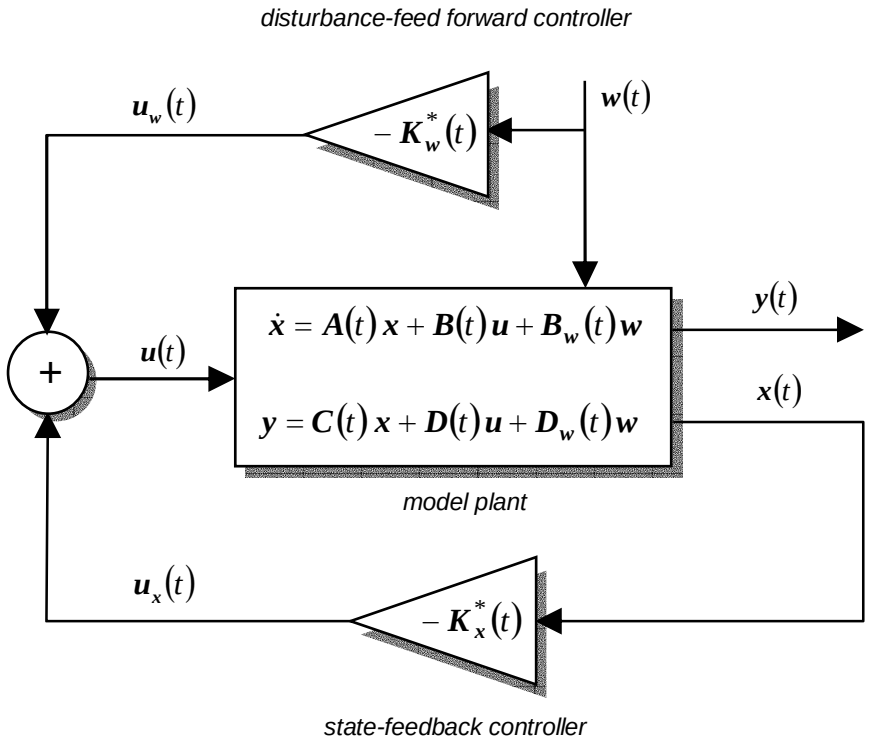

Figure 4.2: Optimal control structure for linear systems with measurable disturbances

4.3 Application of LQR Control to the Spatial Car Model

The linear time-invariant spatial car model (3.78) results from constant velocity v and track curvature κ is used as an example to design the optimal control based on LQR approach. Since the lateral acceleration a_y as disturbance of the system can be measured online by an acceleration sensor, the solution of the LQR problem for linear systems with measurable disturbances introduced in the above section can be applied. To estimate the dynamic characteristics of vehicles in yaw motion, automotive performance criteria specified for the spatial car model are introduced. Advantages and drawbacks of the LQR control design applied to the spatial car model with active suspensions will be shown.

4.3.1 Dynamic criteria for the spatial car model

During cornering, the vehicle weight is transferred between the wheels resulting in load changes. Vehicle ride safety and road handling characteristics, however, are determined by the dynamic wheel loads. High loads allow greater longitudinal and lateral transmission forces between wheels and ground. On the contrary, low wheel loads can cause loss of controllability of the car. Larger magnitudes of the roll and pitch angles will indirectly influence ride safety as contact force magnitudes might reach zero causing lift-off, Sampson [67]. The most commonly used criterion for ride safety of passenger cars is the integral function of quadratic dynamic wheel loads, Mitschke [50], Popp and Schiehlen [59]. Since the dynamic wheel loads are proportional to the suspension deflections, the integral of quadratic suspension deflections can be used as a substitutive criterion, Bestle [7].

For the spatial car model, the four wheels of the car are assumed to be mass-less points which keep their traces on the plane track surface. The dynamic wheel loads are, therefore, proportional to the vertical displacements of the suspension nodes z_{Si}, i = 1(1)4, and thus can be used to represent the *ride safety criterion*, Ammon [2]:

$$f_1 = \sqrt{\frac{1}{4T}\int_0^T \sum_{i=1}^{4} z_{Si}^2(t)\,dt} \quad . \tag{4.47}$$

Ride comfort is mainly characterized by the accelerations of the driver and passengers, Mitschke [50], Popp and Schiehlen [59]. For simple models, the accelerations of the car body can be used alternatively to estimate vehicle ride comfort. While the vertical acceleration is of special interest for cars riding on uneven roads, the roll acceleration of the car body must be taken into consideration for cars in yaw motion on even roads. The *ride comfort criterion* for the spatial car model can then be defined by

$$f_2 = \sqrt{\frac{1}{T}\int_0^T \ddot{\beta}(t)^2\, dt} \quad . \tag{4.48}$$

For passenger cars using active suspensions, improving ride safety and ride comfort often requires a large expense of control energy. Control effort, therefore, must be considered as an objective to be minimized for passenger cars using active suspensions. The *control effort criterion* for the spatial car model can be expressed by

$$f_3 = \sqrt{\frac{1}{4T}\int_0^T \sum_{i=1}^{4} u_i(t)^2\, dt} \quad . \tag{4.49}$$

The formulas (4.47) – (4.49) represent root-mean-squares (RMS/r.m.s) of suspension displacements, car body roll acceleration and control forces, respectively. It should be noted that minimizing the integrals in (4.47) – (4.49) will minimize f_1, f_2 and f_3, respectively. These integrals, therefore, can be also treated as corresponding criteria and will be used alternatively to define the objective function for LQR control design.

4.3.2 Spatial car model simulation

As illustrated in Figure 4.3, the car may move along a circular track of radius R = 10 [m] with constant velocity v = 30 km/h ≈ 8.33 m/s. At the starting point, the lateral acceleration of the car jumps to a constant value $a_y = v^2/R \approx 6.94\ m/s^2$ as a step function exciting the system. The car parameters are given in Appendix A.

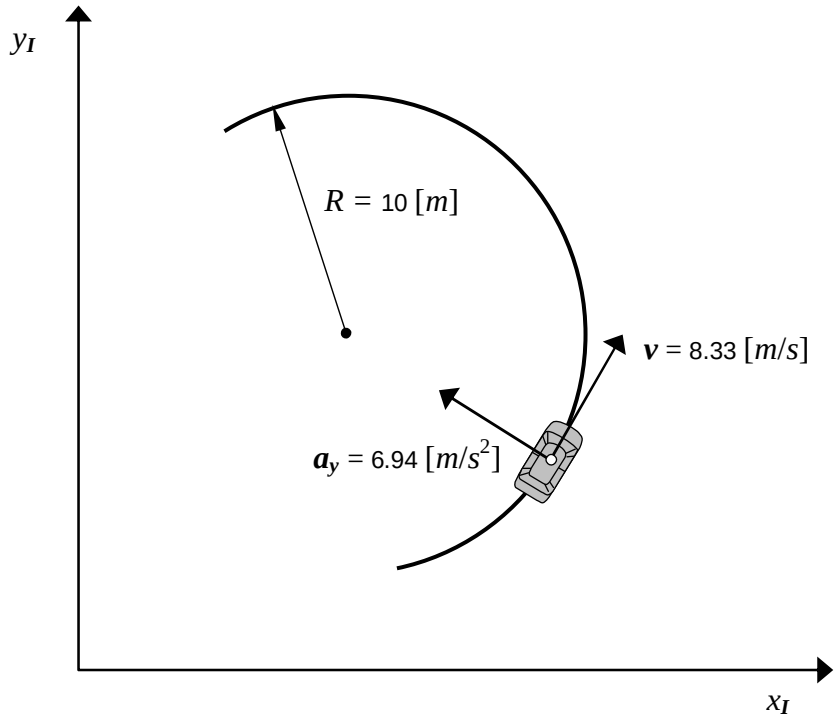

Figure 4.3: Simulation of the spatial car model in cornering

4.3.3 LQR design for the spatial car model

In order to apply the LQR control to the spatial car model, the objective function must be determined first. The quadratic objective function for the spatial car model can be defined as weighted sum of the integrals in (4.47) – (4.49)

$$J = w_1 \int_0^\infty \sum_{i=1}^{4} z_{Si}^2 \, dt + w_2 \int_0^\infty \ddot{\beta}^2 \, dt + w_3 \int_0^\infty \sum_{i=1}^{4} u_i^2 \, dt \ . \tag{4.50}$$

With the vectors of control inputs $\boldsymbol{u}$ and measured outputs $\boldsymbol{y}$ defined by (3.64) and (3.75), respectively, the objective function (4.50) can be expressed in the regular form of the LQR problem (4.3) by introducing the weighting matrices

$$\boldsymbol{Q}_y = diag\left\{ w_1,\ \ w_1,\ \ w_1,\ \ w_1,\ \ w_2 \right\}$$

and

$$\boldsymbol{R}_u = diag\left\{ w_3,\ \ w_3,\ \ w_3,\ \ w_3 \right\} \tag{4.51}$$

where w_1, w_2 and w_3 are the weighting factors on ride safety, ride comfort and control effort, respectively. Once the weighting matrices $\boldsymbol{Q}_y$ and $\boldsymbol{R}_u$ are determined, the MATLAB function '*lqry*' will give the Riccati matrix $\boldsymbol{P}$ and the optimal state-feedback control gain $\boldsymbol{K}_x^*$. The optimal disturbance-feed forward control gain $\boldsymbol{K}_w^*$ can be then easily derived from (4.45) with the matrices $\boldsymbol{R}, \boldsymbol{N}_{uw}^{\boldsymbol{T}}$ and $\boldsymbol{N}_{xw}$ defined in Equation (4.29). By choosing proper weighting factors, desired solutions can be obtained.

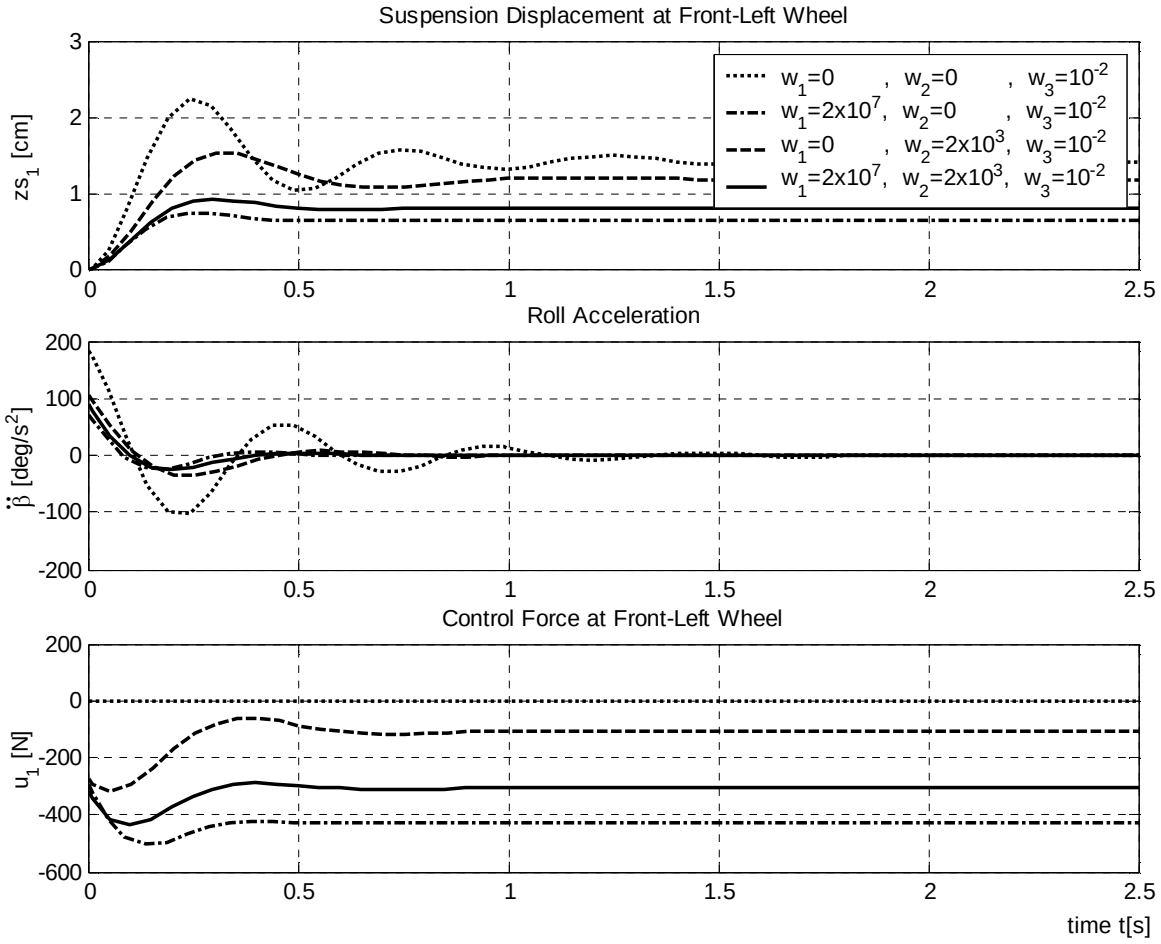

Figure 4.4: Simulation results obtained from LQR design

Figure 4.4 shows the simulation results corresponding to different values of the weighting factors. During simulation the system takes on the form of the linear-time invariant system with constant disturbance $w = a_y$ (3.78). The resulting optimal control gains $\boldsymbol{K}_x^* \in \mathbb{R}^{4\times 6}$ and $\boldsymbol{K}_w^* \in \mathbb{R}^{4\times 1}$ are given in Table 4.1. For clearance, only the time response of the vertical displacement of the suspension node z_{S1} and the control force u_1 at the front-left wheel are plotted in the figure. In case 1 denoted by the dotted lines, only the control effort criterion is minimized resulting in $\boldsymbol{K}_x^*, \boldsymbol{K}_w^* = \boldsymbol{0}$ which means that no control energy is provided by the actuators. The dash-dotted and dashed lines corresponding to case 2 and case 3 represent the responses of the system when the ride safety and ride comfort criterion are optimized separately, respectively. In the last case described by the solid lines all three criteria are penalized and a compromise solution is derived.

Table 4.1: Optimal state-feedback gain matrix $\boldsymbol{K}_x^*$ and disturbance-feed forward gain vector $\boldsymbol{K}_w^*$ corresponding to different weighting factors

case	weighting factors			optimal control gains	
	w_1	w_2	w_3	$\boldsymbol{K}_x^*$ $(\text{x } 10^4)$	$\boldsymbol{K}_w^*$
1	0	0	10^{-2}	$\boldsymbol{0}$	$\boldsymbol{0}$
2	$2\text{x}10^7$	0	10^{-2}	$\begin{bmatrix} 2.8980 & 1.2132 & -3.0599 & 0.4091 & 0.1508 & -0.3908 \\ 2.9099 & -1.3957 & -2.7736 & 0.4084 & -0.1496 & -0.3883 \\ 3.0438 & 1.3990 & 5.3511 & 0.2688 & 0.1484 & 0.4597 \\ 3.0555 & -1.1892 & 5.6352 & 0.2681 & -0.1496 & 0.4622 \end{bmatrix}$	$\begin{bmatrix} 42.1406 \\ -74.7698 \\ 70.7979 \\ -45.1908 \end{bmatrix}$
3	0	$2\text{x}10^3$	10^{-2}	$\begin{bmatrix} 0.0007 & -1.0378 & 0.0415 & 0.0010 & 0.0856 & 0.0155 \\ 0.0006 & 1.1796 & -0.0379 & -0.0008 & -0.0850 & -0.0169 \\ -0.0007 & -1.1814 & 0.0395 & 0.0009 & 0.0872 & 0.0171 \\ -0.0008 & 1.0185 & -0.0393 & -0.0009 & -0.0821 & -0.0150 \end{bmatrix}$	$\begin{bmatrix} 39.8123 \\ -41.8481 \\ 41.6755 \\ -39.3411 \end{bmatrix}$
4	$2\text{x}10^7$	$2\text{x}10^3$	10^{-2}	$\begin{bmatrix} 2.9035 & -0.2623 & -2.9627 & 0.4094 & 0.1674 & -0.3761 \\ 2.9049 & 0.1732 & -2.8718 & 0.4082 & -0.1640 & -0.4036 \\ 3.0487 & -0.1685 & 5.4497 & 0.2690 & 0.1634 & 0.4750 \\ 3.0501 & 0.2636 & 5.5398 & 0.2679 & -0.1654 & 0.4478 \end{bmatrix}$	$\begin{bmatrix} 45.8808 \\ -52.2497 \\ 51.0035 \\ -46.3534 \end{bmatrix}$

The simulation results show the effectiveness of active suspension systems with LQR control on ride safety and ride comfort of the car in cornering and the usefulness of the LQR design as the optimal gain matrices with a large number of components can be easily obtained by choosing appropriate weighting factors. However, if the control problem is subject to constraints on the states, measured outputs or control inputs, the optimal solutions may not be found by choosing the weighting factors by hand.

In order to overcome the drawbacks of the LQR approach in the optimal controller design problem, multi-criterion optimization must be applied. The combination of multi-criterion optimization methods and the LQR algorithm for defining the optimal controllers for the spatial car model will be introduced in the next chapter.

Chapter 5

Multi-criterion Optimization

In technical applications, multiple goals have to be taken into account which often are in conflict. *Multi-criterion optimization* (MCO) is a powerful tool for finding the best compromise solution balancing the conflicts, and therefore is of great importance in practice, particularly in engineering design. In this chapter, two effective MCO methods for generating the trade-off solutions, namely the *compromise method* and the *recursive knee approach*, will be introduced. MCO problems will be defined for both the *passive* and *active suspension* case. An optimization method combining the MCO and LQR algorithm in order to reduce the number of design variables will be presented. Significant improvements in ride safety and ride comfort for the passive and active suspension also will show the effectiveness of the approach.

5.1 Overview on Multi-criterion Optimization

Multi-criterion optimization (MCO) or vector optimization refers to the process of optimizing simultaneously a collection of objective functions. The general multi-criterion optimization problem is to find a *vector of design variables* $\boldsymbol{p}$, $\boldsymbol{p} \in \mathbb{R}^{\boldsymbol{h}}$, optimizing the *vector of criteria* $\boldsymbol{f}(\boldsymbol{p})$, $\boldsymbol{f} : \mathbb{R}^{\boldsymbol{h}} \rightarrow \mathbb{R}^{\boldsymbol{n}}$, subject to *equality constraints* $\boldsymbol{g}(\boldsymbol{p}) = \boldsymbol{0}$, *inequality constraints* $\boldsymbol{h}(\boldsymbol{p}) \leq \boldsymbol{0}$ and *variable bounds* $\boldsymbol{p}^{\boldsymbol{l}}$ and $\boldsymbol{p}^{\boldsymbol{u}}$:

$$\underset{\boldsymbol{p} \in \mathcal{P}}{opt} \quad \boldsymbol{f}(\boldsymbol{p})$$

where

$$\mathcal{P} := \left\{ \boldsymbol{p} \in \mathbb{R}^{\boldsymbol{h}} \;\middle|\; \boldsymbol{g}(\boldsymbol{p}) = \boldsymbol{0}, \boldsymbol{h}(\boldsymbol{p}) \leq \boldsymbol{0}, \boldsymbol{p}^{\boldsymbol{l}} \leq \boldsymbol{p} \leq \boldsymbol{p}^{\boldsymbol{u}}, \boldsymbol{g} : \mathbb{R}^{\boldsymbol{h}} \rightarrow \mathbb{R}^{\boldsymbol{l}}, \boldsymbol{h} : \mathbb{R}^{\boldsymbol{h}} \rightarrow \mathbb{R}^{\boldsymbol{m}} \right\} \quad (5.1)$$

is the *set of admissible designs* or feasible design space, and *opt* means minimization of all *individual criteria* $f_i(\boldsymbol{p})$, $i = 1(1)n$. The criteria define a transformation of the feasible design space $\mathcal{P}$ to the *attainable criterion space* $\mathcal{F}$,

$$\mathcal{F} := \left\{ \boldsymbol{f}(\boldsymbol{p}) \in \mathbb{R}^{\boldsymbol{n}} \;\middle|\; \boldsymbol{p} \in \mathcal{P} \right\}. \tag{5.2}$$

Figure 5.1 illustrates the mapping of the feasible design space $\mathcal{P}$ to the attainable criterion space $\mathcal{F}$ for a bi-criterion case.

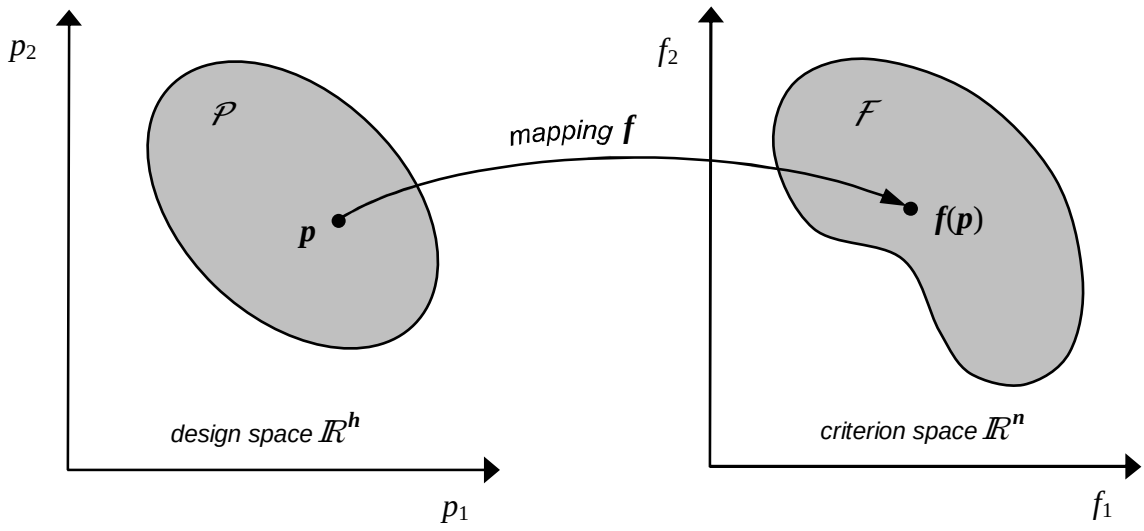

Figure 5.1: Mapping from design space $\mathcal{P} \subset \mathbb{R}^2$ into criteria space $\mathcal{F} \subset \mathbb{R}^2$

If we have only one single objective, i.e. $n = 1$, the problem is a *scalar optimization* problem where the scalar objective defines a *total order* on $\mathcal{P}$. This means all design points $\boldsymbol{p} \in \mathcal{P}$ are comparable to each other. In this case, a design point $\boldsymbol{p}^*$ is called a *global minimum* if

$$f(\boldsymbol{p}^*) \leq f(\boldsymbol{p}) \quad \forall\, \boldsymbol{p} \in \mathcal{P}, \tag{5.3}$$

and called a *local minimum*, if there exits an open set $\mathcal{U} \subset \mathbb{R}^2$ around $\boldsymbol{p}^*$ such that

$$f(\boldsymbol{p}^*) \leq f(\boldsymbol{p}) \quad \forall\, \boldsymbol{p} \in \mathcal{U} \cap \mathcal{P}. \tag{5.4}$$

In real engineering applications usually more than one criterion has to be minimized simultaneously, i.e. $n > 1$, resulting in a *multi-criterion optimization* (MCO) problem. The vector of criteria defines only a *partial order* on $\mathcal{P}$, i.e. not all designs in $\mathcal{P}$ are comparable to each other, Bestle [7]. Since it is impossible to find a design point $\boldsymbol{p}^*$ where all criteria are minimized at the same time, i.e.

$$f_i(\boldsymbol{p}^*) \le f_i(\boldsymbol{p}) \quad \forall\, i\,, \boldsymbol{p} \in \mathcal{P}\,, \tag{5.5}$$

a new concept of optimality, namely *Edgeworth-Pareto optimality*, is defined as follows:

> A design $\boldsymbol{p}^{EP} \in \mathcal{P}$ is called *Edgeworth-Pareto* (EP) *optimal* if there exits no feasible point $\boldsymbol{p} \in \mathcal{P}$ such that $f_i(\boldsymbol{p}) \le f_i(\boldsymbol{p}^{EP})\ \forall\, i$ and $f_j(\boldsymbol{p}) < f_j(\boldsymbol{p}^{EP})$ for at least one j. The design points satisfying this property belong to the *EP-optimal set*
>
> $$\mathcal{P}^{EP} := \left\{ \boldsymbol{p}^{EP} \in \mathcal{P} \;\middle|\; \nexists\, \boldsymbol{p} \in \mathcal{P} : \boldsymbol{f}(\boldsymbol{p}) < \boldsymbol{f}(\boldsymbol{p}^{EP}) \right\} \tag{5.6}$$
>
> where $\boldsymbol{f}(\boldsymbol{p}) < \boldsymbol{f}(\boldsymbol{p}^{EP})$ means $\left(f_i(\boldsymbol{p}) \le f_i(\boldsymbol{p}^{EP})\ \forall\ i \right) \wedge \left(\boldsymbol{f}(\boldsymbol{p}) \ne \boldsymbol{f}(\boldsymbol{p}^{EP}) \right)$.

The set in $\mathbb{R}^n$ formed by the criteria vectors of the EP-optimal solutions

$$\mathcal{F}^{EP} := \boldsymbol{f}(\mathcal{P}^{EP}) \tag{5.7}$$

is always located on the boundary of the attainable criterion space $\mathcal{F}$, and therefore it is known as the *Pareto frontier* or *trade-off curve*, whereas the Pareto-optimal set $\mathcal{P}^{EP}$ has not necessarily to lie on the boundary of the feasible design space $\mathcal{P}$ as illustrated in Figure 5.2.

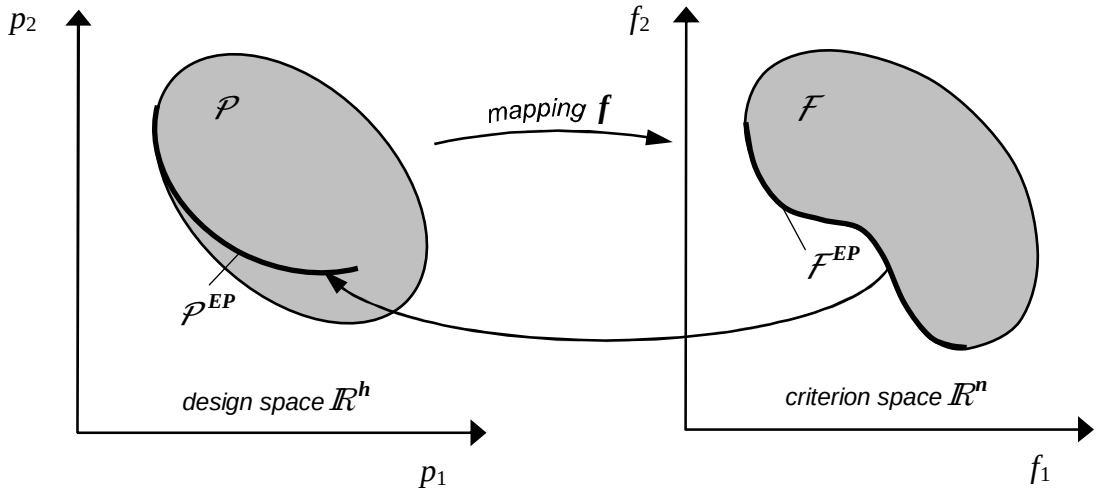

Figure 5.2: Pareto-optimal set $\mathcal{P}^{EP}$ and Pareto frontier $\mathcal{F}^{EP}$

The end points F_i^* of the Pareto frontier are called *individual minima*. They are defined by vectors $\boldsymbol{f}_i^* := [\, f_1(\boldsymbol{p}_i^*), \ldots, f_i^*, \ldots, f_n(\boldsymbol{p}_i^*)]^T$ where $\boldsymbol{p}_i^*$ denotes the *individual optimizer* of the criterion $f_i(\boldsymbol{p})$ with *minimum* f_i^*, i.e.

$$f_i^* := \min_{\boldsymbol{p} \in \mathcal{P}} f_i(\boldsymbol{p}), \quad i = 1(1)n\,. \tag{5.8}$$

The hyper-plane running through all individual minima is known as the *convex hull of the individual minima* (CHIM), Das and Denis [19]. For the bi-criterion problem, the CHIM is the line joining the individual minima F_1^* and F_2^*, which are respectively defined by vectors:

$$\boldsymbol{f}_1^* = \begin{bmatrix} f_1(\boldsymbol{p}_1^*) \\ f_2(\boldsymbol{p}_1^*) \end{bmatrix} =: \begin{bmatrix} f_1^* \\ f_2^{1*} \end{bmatrix}, \qquad \boldsymbol{f}_2^* = \begin{bmatrix} f_1(\boldsymbol{p}_2^*) \\ f_2(\boldsymbol{p}_2^*) \end{bmatrix} =: \begin{bmatrix} f_1^{2*} \\ f_2^* \end{bmatrix},$$

see Figure 5.3.

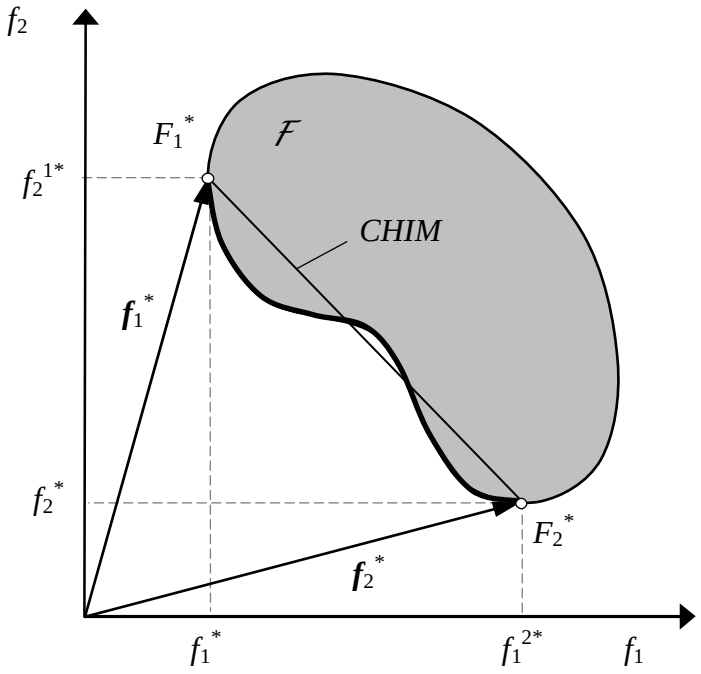

Figure 5.3: Individual minima F_1^* and F_2^* and CHIM

5.2 Multi-criterion Optimization Methods

In general, EP-optimal solutions are not unique, thus the designer has to choose a single EP-optimal point from the set $\mathcal{P}^{EP}$ as the desired solution instead. To find such points or a representative subset of $\mathcal{P}^{EP}$, the multi-criterion optimization problem usually has to be reduced to scalar utility problems. In this section, two scalarization approaches applied to the MCO problem for the spatial car model for obtaining a sample set of points of the Pareto frontier, namely the *compromise method* and the *recursive knee approach*, are introduced. Other MCO methods can be found in Bestle [7], Collette and Siarry [16] and Deb [20].

5.2.1 Compromise method

The compromise method entails minimization of one of the criteria $f_r(\boldsymbol{p})$, while expressing the remaining criteria in the form of inequality constraints:

$$\min_{\boldsymbol{p} \in \mathcal{P}^r} f_r(\boldsymbol{p}) \quad \textit{where} \quad \mathcal{P}^r := \left\{ \boldsymbol{p} \in \mathcal{P} \;\middle|\; f_i(\boldsymbol{p}) \le \varepsilon_i,\ i \in \mathcal{I}_n \setminus \{r\} \right\}. \tag{5.9}$$

In the above formulation, ε_i represents an upper bound for f_i and can be considered as scalarizing parameter, Das [17, 18]. By progressively changing the constraint values ε_i, different points on the Pareto frontier can be sampled.

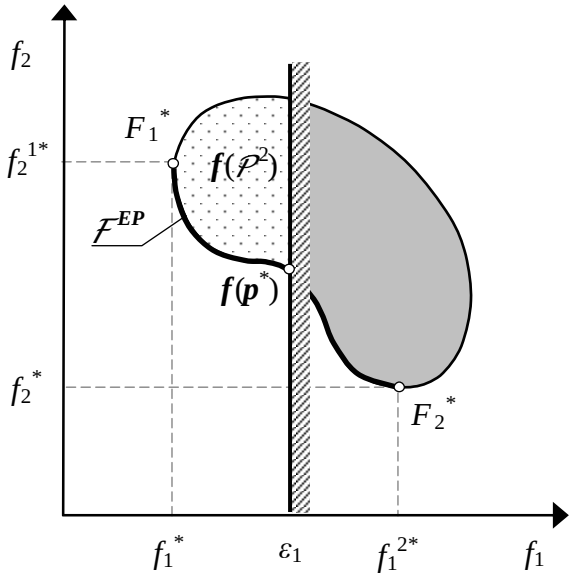

Figure 5.4: Compromise method for the bi-criterion problem with $r = 2$

Figure 5.4 visualizes the working of the compromise method for a bi-criterion case, where f_2 is retained as a criterion, while f_1 is treated as a constraint $f_1 \leq \varepsilon_1$. With this constraint, the original attainable criterion space $\mathcal{F}$ is reduced to $\mathcal{F}^2 := \boldsymbol{f}(\mathcal{P}^2)$, which is the left-upper portion of $\mathcal{F}$ bounded by ε_1. The solution of the problem (5.9) depends strongly on the value of the constraint ε_1. As can be seen from the figure, if $\varepsilon_1 < f_1^* := f_1(\boldsymbol{p}_1^*)$ is chosen, there exits no feasible solution to the stated problem. On the other hand, if $\varepsilon_1 > f_1^{2*} := f_1(\boldsymbol{p}_2^*)$ is used, the entire search space is feasible and the resulting problem finds the solution point F_2^*.

A remarkable advantage of the compromise method is that the EP-optimal solutions can be found even if the criterion space is non-convex. Moreover, these solutions can be distributed evenly on the Pareto frontier by changing appropriately the values for ε_i within the minimum and maximum value of the individual criterion f_i. For the bi-criterion case as illustrated in Figure 5.4, N EP-optimal points on the Pareto frontier $\mathcal{F}^{EP}$ between the individual minima F_1^* and F_2^* can be found by choosing the constraint values of ε_1 as

$$\varepsilon_{1k} = f_1^{2*} - k\,\Delta\varepsilon_1, \qquad k = 1(1)N \tag{5.10}$$

where

$$\Delta\varepsilon_1 := \frac{f_1^{2*} - f_1^*}{N+1}, \quad f_1^* := f_1(\boldsymbol{p}_1^*), \quad f_1^{2*} := f_1(\boldsymbol{p}_2^*).$$

This formulation means that ε_1 is reduced evenly from f_1^{2*} to f_1^* with the constant difference $\Delta\varepsilon_1$.

The effectiveness of the compromise method is, however, limited to problems with not more than two criteria. For problems with more than two criteria, values of the constraints for more than one criterion have to be determined in advance. Many sets of the constraint values need to be chosen and the corresponding number of associated optimization problems increases. This results in wasted computational resources as will be shown in Section 5.4, where the compromise method is applied to the three-criterion optimization problem defined for the spatial car model.

5.2.2 Recursive knee approach

The recursive knee approach is a technique based on the *normal-boundary intersection* (NBI) method introduced first by Das and Denis [19]. This technique was then modified and programmed by Wachal and Bestle [86], whose main idea is presented below.

The NBI sub-problem maximizes the distance along a normal while staying feasible in an attempt to find the point of intersection between the normal and the Pareto frontier. Let $\boldsymbol{F}$ be the matrix defined as $\boldsymbol{F} := [\ \boldsymbol{f}_1^*,\ \boldsymbol{f}_2^*, \ldots,\ \boldsymbol{f}_n^*\ ]$. Then an arbitrary point $\boldsymbol{A}$ on the CHIM can be determined by vector $\boldsymbol{r}_A := \boldsymbol{Fa}$, where $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{R}^n$ is treated as scalarizing parameter with

$$\mathcal{A} := \left\{ \boldsymbol{a} \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} a_i = 1,\ a_i \geq 0 \right\}. \tag{5.11}$$

At a given point $\boldsymbol{A}$, define a vector $\boldsymbol{n}$ normal to the CHIM pointing away from the origin. Then, the NBI sub-problem for obtaining a single EP-optimal solution can be formulated as follows

$$\begin{aligned} &\min_{\boldsymbol{p} \in \mathcal{P},\, t} \quad t \\ &\textit{subject to} \quad \boldsymbol{Fa} + t\boldsymbol{n} = \boldsymbol{f}(\boldsymbol{p}), \\ &\qquad\qquad \boldsymbol{a} \in \mathcal{A}\,. \end{aligned} \tag{5.12}$$

The NBI algorithm for a bi-criterion case is illustrated in Figure 5.5. For finding a representative subset of solutions, point $\boldsymbol{A}$ has to be moved along the CHIM, and thus the value of vector $\boldsymbol{a}$ has to be varied. A systematic method of setting $\boldsymbol{a}$ in order to generate an evenly distributed set of EP-optimal points on the Pareto frontier can be found in Das and Denis [19].

If the origin point of the line $\boldsymbol{Fa} + t\boldsymbol{n}$, i.e. point $\boldsymbol{A}$, is allowed to be an optimization variable itself, the solution to the problem yields the EP-solution that is furthest from the CHIM in normal direction. The NBI sub-problem (5.12) therefore has to be reformulated with $\boldsymbol{a}$ becoming a design variable as well:

$$\begin{aligned} &\min_{\boldsymbol{p}\in\mathcal{P},\,t,\,\boldsymbol{a}} \quad t \\ &\textit{subject to} \quad \boldsymbol{F}\boldsymbol{a} + t\boldsymbol{n} = \boldsymbol{f}(\boldsymbol{p}), \\ &\qquad\qquad \boldsymbol{a}\in\mathcal{A}. \end{aligned} \tag{5.13}$$

This sub-problem is called the *knee sub-problem*, and the solution to this sub-problem characterizes the *knee of the Pareto frontier*.

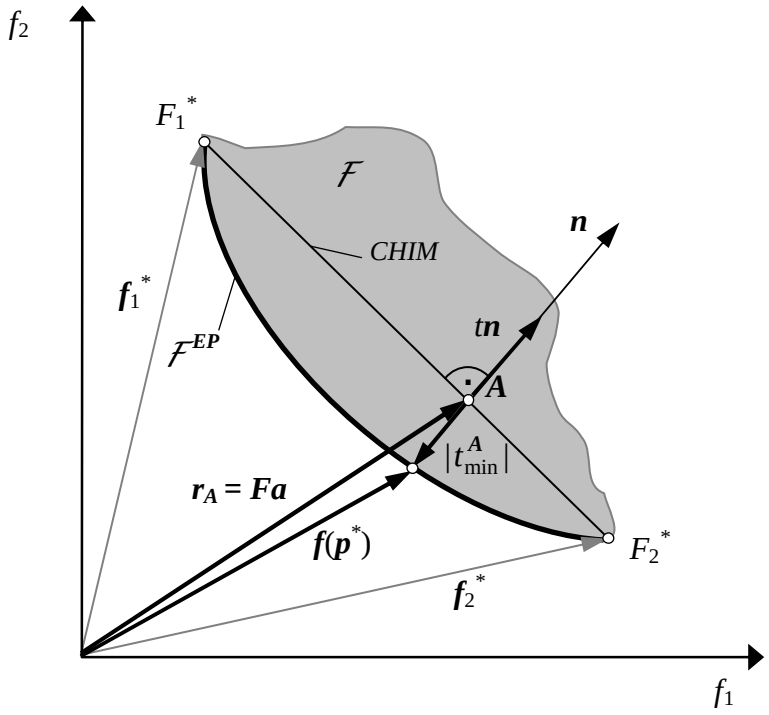

Figure 5.5: Normal-boundary intersection (NBI) method

Once the knee of the Pareto frontier is obtained, the CHIM can be refined by considering the piece-wise linear approximation joining the individual minima with the knee, as shown in Figure 5.6. Each linear segment can be considered as a separate sub-CHIM, and the knee sub-problem can be solved for each sub-CHIM. The process can be repeated recursively to generate several points on the Pareto frontier.

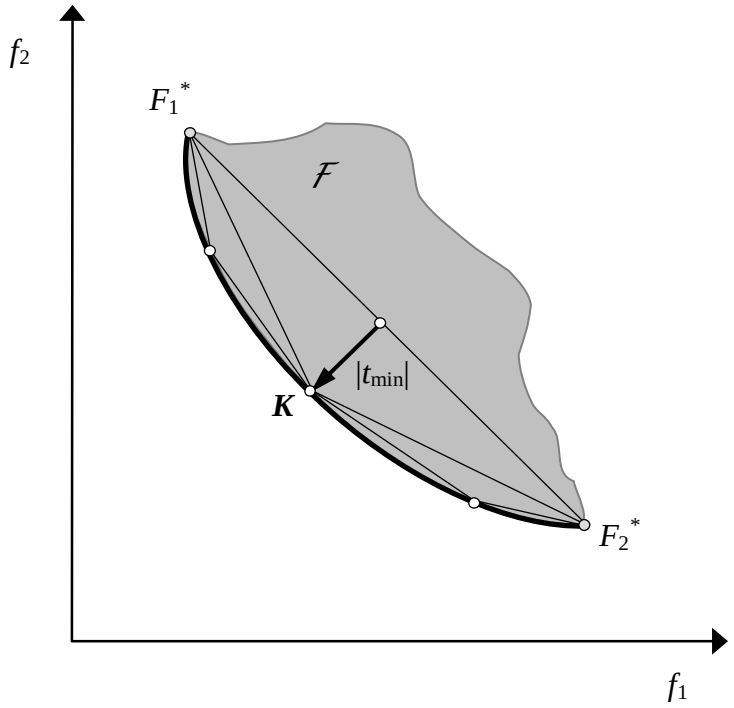

Figure 5.6: Recursive knee approach

Since the solution to the knee sub-problem yields the point that is 'the furthest' from the piece-wise linear approximation, every solution to a subsequent knee sub-problem yields the best refinement of the approximation possible. This procedure results in building a good approximation to the Pareto frontier by solving a minimal number of computationally intensive sub-problems.

Since the origin point of the normal direction is a variable in the knee sub-problem, the disconnected nature of the Pareto frontier poses no infeasibility problem for the technique as it did for the traditional NBI method. The *recursive knee approach* can easily be applied to problems with more than two criteria as well. Since the approach finds the EP-solutions that would best improve the piece-wise linear approximation, it does not miss important parts of the Pareto frontier, Das [18].

The pre-eminence of the recursive knee approach over the compromise method to the three-criterion optimization problem of the spatial car model will be shown in Section 5.5.

5.3 MCO Problem for Passive Suspension

In this section, a *bi-criterion* optimization problem for the spatial car model with a *passive suspension* system will be defined and solved with the *compromise method*. The significant improvements in both car ride safety and ride comfort will be shown.

5.3.1 Problem definition

For the case of a passive suspension, the two objectives to be minimized are *ride safety* (4.47) and *ride comfort* (4.48). The chosen design parameters are the damping coefficients b_f and b_r of the front and rear suspensions, the stiffnesses k_f and k_r of the front and rear suspensions, and the modified rotational stiffnesses r_f and r_r of the front and rear anti-roll bars. The MCO problem for the spatial car model with a passive suspension then reads as

$$\min_{\boldsymbol{p} \in \mathcal{P}} \boldsymbol{f}(\boldsymbol{p}), \qquad \boldsymbol{f}(\boldsymbol{p}) := [\, f_1,\ f_2 \,]^{\boldsymbol{T}},$$

$$\mathcal{P} := \left\{ \boldsymbol{p} = \left[b_f,\ b_r,\ k_f,\ k_r,\ r_f,\ r_r \right]^{\boldsymbol{T}} \in \mathbb{R}^6 \;\middle|\; \boldsymbol{p}^{\boldsymbol{l}} \le \boldsymbol{p} \le \boldsymbol{p}^{\boldsymbol{u}} \right\}$$

where

$$\boldsymbol{p}^{\boldsymbol{l}} = [1000,\ 1000,\ 15000,\ 15000,\ 15000,\ 5000]^{\boldsymbol{T}},$$

$$\boldsymbol{p}^{\boldsymbol{u}} = [2000,\ 2000,\ 25000,\ 25000,\ 25000,\ 15000]^{\boldsymbol{T}}. \qquad (5.14)$$

5.3.2 Optimization results based on the compromise method

To apply the compromise method, the comfort criterion f_2 is chosen to be minimized, while the ride safety criterion f_1 is expressed as inequality constraint. The MCO problem for the passive suspension case (5.14) then reduces to

$$\min_{\boldsymbol{p} \in \mathcal{P}^2} f_2(\boldsymbol{p}) \quad \textit{where} \quad \mathcal{P}^2 := \left\{ \boldsymbol{p} \in \mathcal{P} \;\middle|\; f_1(\boldsymbol{p}) \le \varepsilon_1 \right\}. \qquad (5.15)$$

with properly chosen upper bounds ε_1 on ride safety. As indicated earlier in Section 5.2.1, to ensure that all the EP-optimal solutions of the problem will be found, the value of the constraint ε_1 must be chosen within the values f_1^* and f_1^{2*} of the ride safety criterion f_1. Thus, these values must be computed first by solving the single optimization problems (5.8). Once the individual minima F_1^* and F_2^* are determined, N EP-optimal points between F_1^* and F_2^* can be generated evenly on the Pareto frontier by solving progressively the problem (5.15) for N values of the constraint ε_1 defined by (5.10).

The obtained optimization results for the spatial car model with passive suspension based on the compromise method are presented in Table 5.1 and shown in Figure 5.7. The simulation parameters have been the same as the ones used for the LQR design in Section 4.3.2, i.e. v = 30 *km/h* and κ = 1/10 m^{-1}.

Table 5.1: EP-optimal solutions for passive suspension optimization

point	criteria		optimized design variables					
	f_1	f_2	b_f	b_r	k_f	k_r	r_f	r_r
	$[cm]$	$[deg/s^2]$	x10^3 $[Ns/m]$		x10^4 $[N/m]$		x10^4 $[N/m/rad]$	
1	1.4905	47.5780	2.0000	2.0000	1.9952	1.7491	1.9195	0.9593
2	1.4651	47.6340	2.0000	1.9998	2.0070	1.7563	1.9367	0.9762
3	1.4394	47.7035	2.0000	1.9983	2.0190	1.7667	1.9544	0.9928
4	1.4132	47.7442	2.0000	2.0000	2.0295	1.7771	1.9723	1.0118
5	1.3882	47.8133	1.9987	2.0000	2.0394	1.7867	1.9913	1.0298
6	1.3629	47.8568	2.0000	2.0000	2.0499	1.7953	2.0119	1.0478
7	1.3373	47.9157	2.0000	2.0000	2.0598	1.8051	2.0324	1.0679
8	1.3127	47.9730	2.0000	2.0000	2.0710	1.8145	2.0498	1.0899
9	1.2868	48.0346	2.0000	2.0000	2.0828	1.8261	2.0741	1.1079
10	1.2586	48.1030	2.0000	2.0000	2.0953	1.8384	2.0984	1.1320
11	1.2338	48.1644	2.0000	2.0000	2.1072	1.8509	2.1206	1.1528
12	1.2098	48.2247	2.0000	2.0000	2.1204	1.8639	2.1403	1.1749
13	1.1831	48.2928	2.0000	2.0000	2.1329	1.8762	2.1654	1.2004
14	1.1578	48.3589	2.0000	2.0000	2.1440	1.8903	2.1896	1.2254
15	1.1312	48.4297	2.0000	2.0000	2.1614	1.9086	2.2146	1.2500
16	1.0999	48.5147	2.0000	2.0000	2.1753	1.9216	2.2492	1.2841
17	1.0799	48.5701	2.0000	2.0000	2.1831	1.9338	2.2687	1.3089
18	1.0540	48.6434	2.0000	2.0000	2.1967	1.9473	2.2986	1.3384
19	1.0304	48.7113	2.0000	2.0000	2.2111	1.9615	2.3261	1.3653
20	1.0026	48.7930	2.0000	2.0000	2.2293	1.9832	2.3557	1.4000
21	0.9770	48.9500	1.9924	2.0000	2.2423	1.9970	2.3889	1.4350

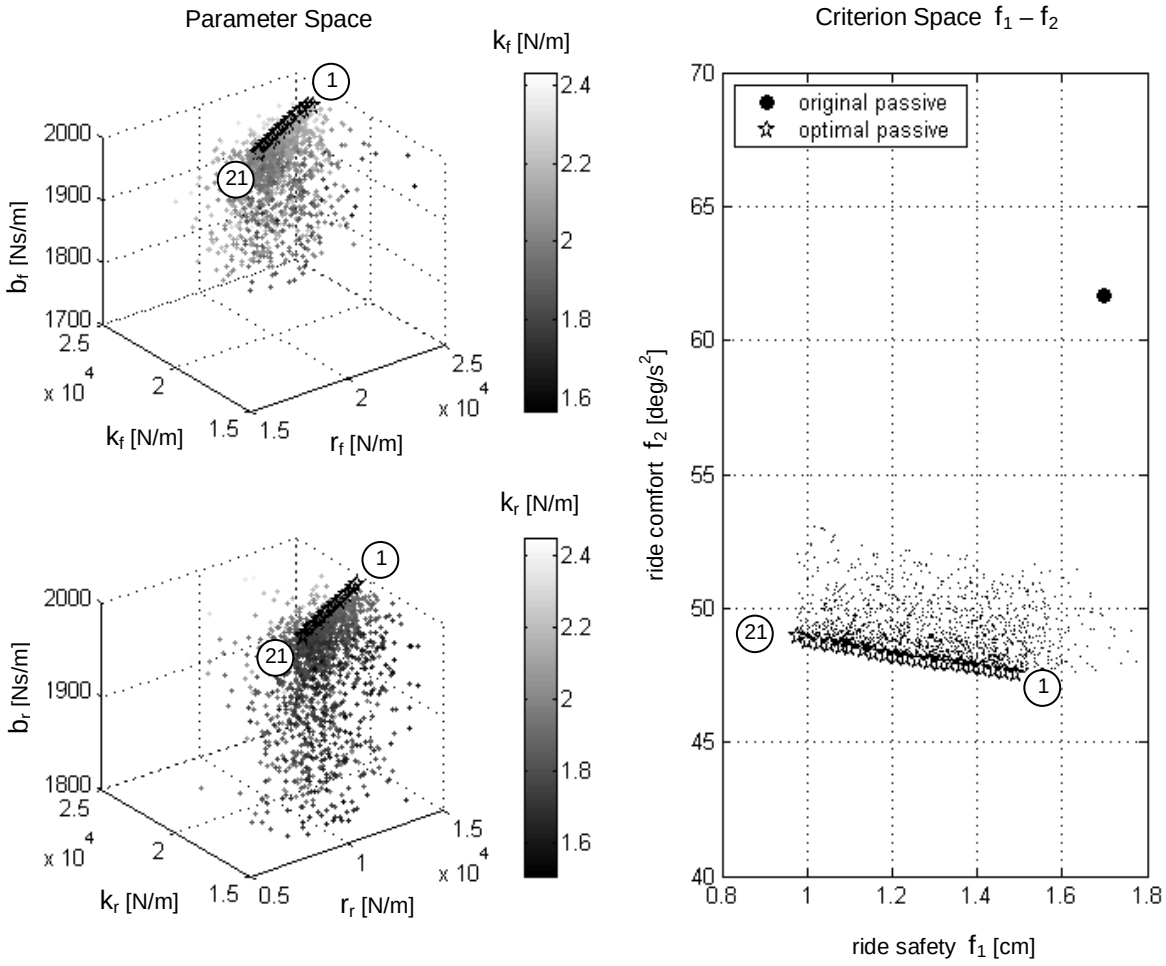

Figure 5.7: Optimization results for the spatial car model with passive suspension

Figure 5.7 shows the EP-optimal solutions in the parameter spaces and the criterion space by the stars. The dots are the solutions for randomly created design parameters to give some impressions about the feasible spaces. As can be seen from the criterion space, an almost even distribution of the EP-optimal solutions on the Pareto frontier has been generated by changing evenly the constraint value. The success of solving the MCO problems depends highly on the starting design points. To converge more quickly, the optimal result of (5.15) for some $\varepsilon_1^{(k-1)}$ was taken as starting point for the optimization problem with the next bound $\varepsilon_1^{(k)}$:

$$\boldsymbol{p}_0^{(k)} = \boldsymbol{p}^{*(k-1)}, \qquad k = 1(1)N, \qquad \boldsymbol{p}^{*(0)} := \boldsymbol{p}_2^{*}. \tag{5.16}$$

Compared to the case of the original passive suspension with parameters given in Appendix A denoted by the black circle in Figure 5.7, the optimized passive suspension improves the car ride safety by about 12% − 42% and ride comfort by 21% − 23%. The obtained results show that improving ride safety in cornering situations requires an increase in both the springs and anti-roll bars stiffness, which is contrary to the case improving ride comfort.

5.4 MCO Problem for Active Suspension

Ride safety and ride comfort of the car can be further improved by using active suspension. In this section, the *three-criterion* optimization problem for the spatial car model with *active suspension* will be defined. An optimization method combining the multi-criterion optimization concept with the linear quadratic regulator (LQR) algorithm will be introduced. The drawback of the compromise method compared to the recursive knee approach for the three-criterion optimization problem will be exposed based on the results obtained from the proposed optimization algorithm.

5.4.1 Problem definition

Different from the passive suspension case, the existence of the control forces generated by actuators in the active suspension system requires control effort which has to be taken into account in an optimization problem. The criteria vector, therefore, includes not only *ride safety* f_1 and *ride comfort* f_2, but also the *control effort criterion* (4.49). The spatial car model is set up the same way as in the case of LQR design and the control structure with state-feedback and disturbance-feed forward controller is according to Figure 4.2. The parameters of the optimized passive suspension corresponding to the 10^{th} row in Table 5.1 are used as constants for the passive suspension parts. The design variables are the components k_{xij} of the 4x6-state-feedback gain matrix $\boldsymbol{K}_x$, and the elements k_{wj} of the 4x1-disturbance-feed forward gain vector $\boldsymbol{K}_w$. The MCO problem for the spatial car model with active suspension can then be stated as

$$\min_{\boldsymbol{p}\in\mathcal{P}} \boldsymbol{f}(\boldsymbol{p}), \qquad \boldsymbol{f}(\boldsymbol{p}) = [\, f_1,\ f_2,\ f_3 \,]^{\boldsymbol{T}},$$

$$\mathcal{P} := \left\{ \boldsymbol{p} = [\, k_{x11},\ ...,\ k_{x46},\ k_{w1},\ ...,\ k_{w4} \,]^{\boldsymbol{T}} \in \mathbb{R}^{28} \,\middle|\, h(\boldsymbol{p}) \le 0 \right\}$$

where

$$h(\boldsymbol{p}) := \max_{i} \left(\max_{t} | u_i(t) | \right) - 500 \ . \tag{5.17}$$

In the above formulation, the design parameters are considered as unbounded. However, an inequality constraint $h(\boldsymbol{p})$ is used to limit the control forces, according to the limited capability of real actuators.

5.4.2 MCO with LQR control

Solving the MCO problems of the form (5.17) directly for the optimal control gains $\boldsymbol{K}_x^*$ and $\boldsymbol{K}_w^*$ without pre-knowledge about the solutions or bounds would be rather time-consuming, especially for large systems like the spatial car model. This problem can be solved faster by applying the LQR algorithm. Instead of finding directly the components of the controllers via optimization, the LQR algorithm is used to compute $\boldsymbol{K}_x^*$ and $\boldsymbol{K}_w^*$ with weighting factors w_1, w_2 and w_3 provided by the optimizer.

With inclusion of LQR control, the MCO problem (5.17) can be reformulated as follows:

$$\min_{\boldsymbol{p}\in\mathcal{P}} \boldsymbol{f}(\boldsymbol{p}), \qquad \boldsymbol{f}(\boldsymbol{p}) = [\, f_1,\ f_2,\ f_3 \,]^{\boldsymbol{T}},$$

$$\mathcal{P} := \left\{ \boldsymbol{p} = [\, w_1,\ w_2,\ w_3 \,]^{\boldsymbol{T}} \in \mathbb{R}^3 \,\middle|\, \boldsymbol{p}^{\boldsymbol{l}} \le \boldsymbol{p} \le \boldsymbol{p}^{\boldsymbol{u}},\ h(\boldsymbol{p}) \le 0 \right\}$$

where

$$\boldsymbol{p}^{\boldsymbol{l}} = \left[\, 0, \quad 0, \quad 10^{-6} \,\right]^{\boldsymbol{T}},$$

$$\boldsymbol{p}^{\boldsymbol{u}} = \left[10^{8}, \quad 10^{8}, \quad 10^{2} \right]^{\boldsymbol{T}},$$

$$h(\boldsymbol{p}) := \max_{i} \left(\max_{t} | u_i(t) | \right) - 500. \tag{5.18}$$

The optimization procedure can be described by Figure 5.8. As illustrated in the figure, the LQR algorithm provides the optimal control gains $\boldsymbol{K}_x^*$ and $\boldsymbol{K}_w^*$ for the spatial car model based on weighting factors obtained from the MCO loop. Then a simulation is performed and the criteria and nonlinear inequality constraint are computed and returned to the MCO loop. The final values of $\boldsymbol{K}_x^*$ and $\boldsymbol{K}_w^*$ corresponding to the optimal weighting factors of the problem (5.18) are the solution of the problem (5.17). By the combination of the LQR algorithm, the number of design variables for the MCO problem defining the optimal controllers can be reduced to the number of criteria.

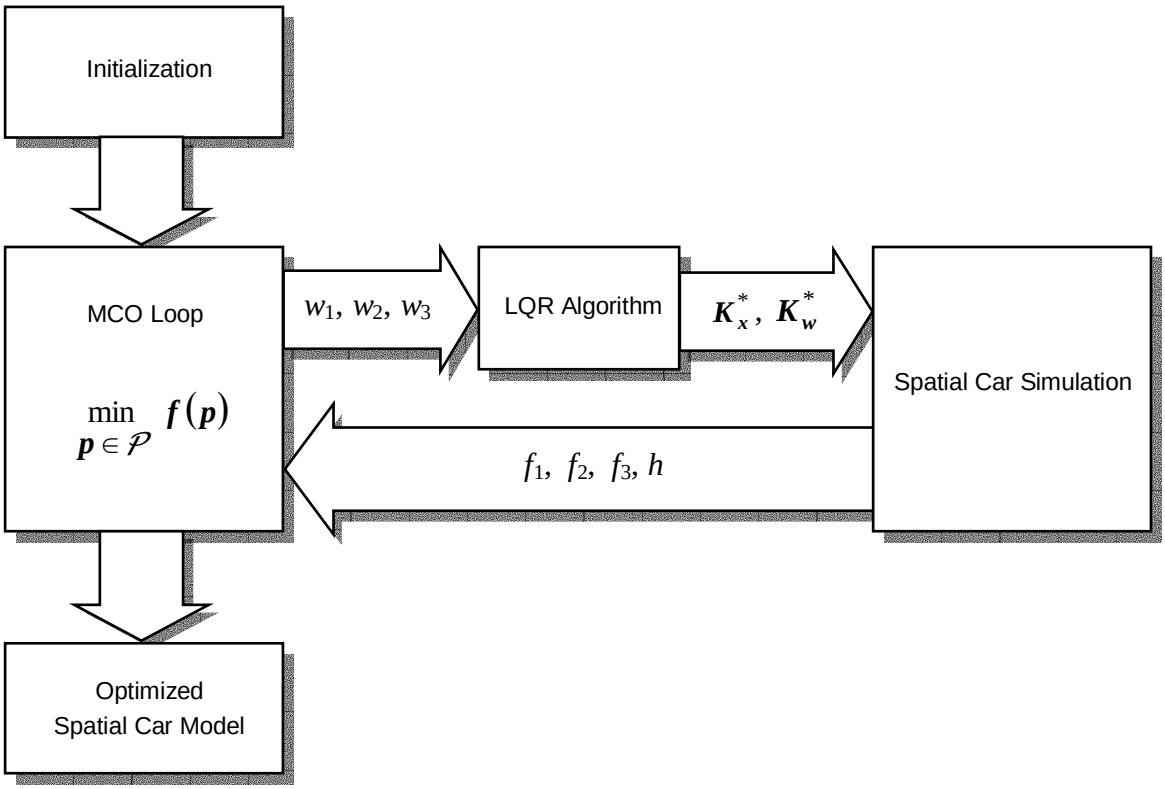

Figure 5.8: Optimization procedure for the spatial car model with active suspension using LQR algorithm

The optimization algorithm using LQR control introduced above is concretized in the applications of the compromise method and recursive knee approach to the MCO problem of the spatial car model (5.18) presented in the sections below.

5.4.3 Optimization results based on the compromise method

Let the ride comfort f_2 be the criterion to be minimized while ride safety f_1 and control effort f_3 are expressed as inequality constraints. The MCO problem for the spatial car model with active suspension (5.18) can then be reduced to the scalar optimization problem

$$\min_{\boldsymbol{p} \in \mathcal{P}^2} f_2(\boldsymbol{p}) \quad \textit{where} \quad \mathcal{P}^2 := \left\{ \boldsymbol{p} \in \mathcal{P} \mid f_1(\boldsymbol{p}) \le \varepsilon_1,\ f_3(\boldsymbol{p}) \le \varepsilon_3 \right\}. \tag{5.19}$$

In order to define proper values for the constraints, the individual minima must be determined first by solving the single optimization problems (5.8). Then, the constraint bound ε_3 can be defined as

$$\varepsilon_3^{(j)} = f_3^{\boldsymbol{u}} - (j-1)\Delta\varepsilon_3 \ , \qquad j = 1(1)M \tag{5.20}$$

where

$$\Delta\varepsilon_3 = \frac{f_3^{\boldsymbol{u}} - f_3^{*}}{M} \ ,$$

$$f_3^{\boldsymbol{u}} := \max\left(f_3^{1*}, f_3^{2*}\right), \quad f_3^{1,2*} := f_3\left(\boldsymbol{p}_{1,2}^{*}\right), \quad f_3^{*} := f_3\left(\boldsymbol{p}_3^{*}\right)$$

and M is a user-defined number of sections. Corresponding to $\varepsilon_3^{(1)}$, N evenly varying values of ε_1 within the individual minima f_1^{*} and f_1^{2*} can be derived by (5.10). To define proper values of ε_1 corresponding to the other values $\varepsilon_3^{(2)}$ to $\varepsilon_3^{(M)}$, sub-individual minima must be defined by solving the following problems:

$$f_{1,2}^{*(j)} := \min_{\boldsymbol{p} \in \mathcal{P}^{3(j)}} f_{1,2}(\boldsymbol{p}), \quad \mathcal{P}^{3(j)} := \left\{ \boldsymbol{p} \in \mathcal{P} \mid f_3(\boldsymbol{p}) \le \varepsilon_3^{(j)} \right\}, \quad j = 2(1)M \ . \tag{5.21}$$

Once the values of the constraints ε_1 and ε_3 are determined, the problem (5.19) can be solved progressively for different fixed values of the constraint ε_3 and associated varying values of the constraint ε_1.

Solutions of the problem (5.19) are shown in Figure 5.9 for different values of the control effort. When the constraint bound on the control effort ε_3 is decreased to $\varepsilon_3 = 0$, the trade-off curves tend to the black star corresponding to the case of the passive suspension with zero-control effort. For the case of no constraint on control effort f_3, i.e. $\varepsilon_3 \to \infty$, the

EP-optimal solutions do not converge to zero, but go to the boundary represented by the six-pointed stars in the figure. This results from the inequality constraint $h(\boldsymbol{p})$ in (5.18) used to limit the maximum value of control forces generated by the actuators.

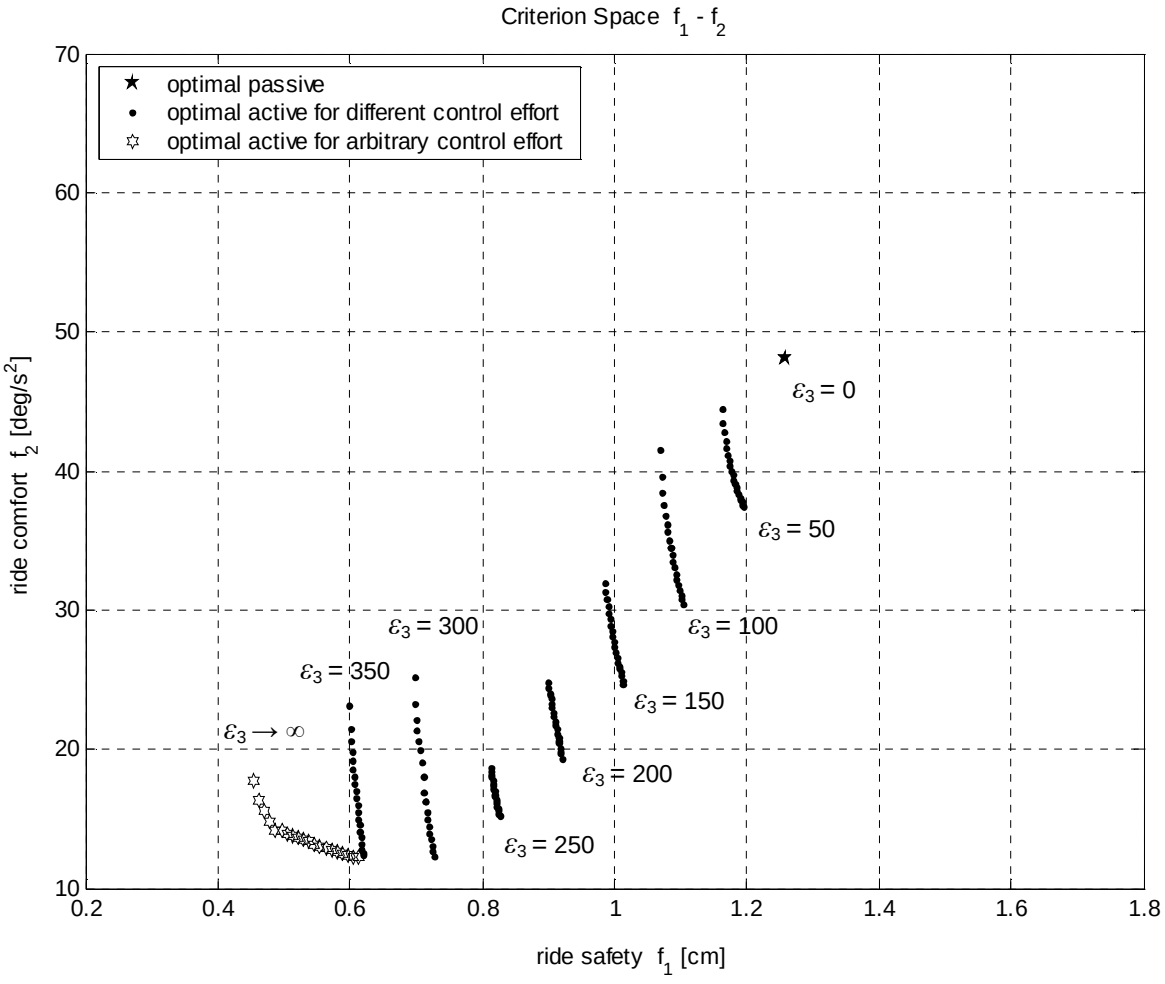

Figure 5.9: Results of the three-criterion optimization problem based on compromise method

As can be seen from the figure, both the car ride safety and ride comfort are significantly improved when the active suspension with state-feedback and disturbance-feed forward control is optimized. These improvements depend on the value of control effort as depicted in Figure 5.9. As shown in Table 5.2, where the EP-optimal solutions for the case of

arbitrary control effort are given, ride safety and ride comfort of the car with optimal active suspension are improved by about 51% − 64% and 63% − 75%, respectively.

Table 5.2: EP-optimal solutions for active suspension optimization based on compromise method for arbitrary control effort

point	criteria			design variables		
	f_1 $[cm]$	f_2 $[deg/s^2]$	f_3 $[N]$	w_1 x10^4	w_2	w_3 x10^{-4}
1	0.6134	12.2040	360.8736	6.9832	20.4031	0.3543
2	0.6055	12.2500	360.8453	6.9832	18.9070	0.3344
3	0.5966	12.3510	361.6310	6.9832	17.5469	0.3163
4	0.5882	12.4710	365.3826	6.9832	16.3046	0.2997
5	0.5798	12.6150	369.2553	6.9832	15.1656	0.2844
6	0.5714	12.7543	373.1473	6.9832	14.1182	0.2703
7	0.5630	12.8940	377.0474	6.9832	13.1524	0.2572
8	0.5547	13.0354	380.9625	6.9832	12.2597	0.2452
9	0.5463	13.1798	384.8833	6.9832	11.4330	0.2340
10	0.5379	13.3274	388.8147	6.9832	10.6657	0.2236
11	0.5295	13.4787	392.7540	6.9832	9.9525	0.2140
12	0.5211	13.6339	396.7014	6.9832	9.2913	0.2051
13	0.5128	13.7932	400.6561	6.9832	8.6756	0.1968
14	0.5044	13.9566	404.6178	6.9832	8.1002	0.1891
15	0.4960	14.1241	408.5850	6.9832	7.5285	0.1816
16	0.4876	14.1787	412.6083	6.9832	6.6386	0.1729
17	0.4792	14.8379	416.1475	6.9832	5.6428	0.1649
18	0.4709	15.6020	419.6442	6.9832	4.8395	0.1594
19	0.4625	16.3784	423.2026	6.9832	4.1990	0.1561
20	0.4541	17.6671	426.5835	6.9832	3.2998	0.1575

The optimization procedure introduced above has shown the complication of the compromise method to a three-criterion optimization problem. To get a figure about solutions, a large number of optimization problems are required to be solved and increases the risk of obtaining more local optima. Then, the problem has to be resolved with one or more new starting points which results in wasted computational time. In order to overcome this drawback of the compromise method, the recursive knee approach may be used instead.

5.5.4 Optimization results based on the recursive knee approach

Based on the optimization algorithm programmed by Wachal and Bestle [86], the recursive knee approach introduced in Section 5.2.2 is applied to solve the MCO problem for the spatial car model with active suspension. The obtained optimization results are shown in Figure 5.10 and some of them are cited in Table 5.3.

Different from the compromise method, the number of EP-optimal solutions for the recursive knee approach can not be defined in à priori since the iterations are terminated according to the knee distance t of the knee sub-problems (5.13), which depend on the form of the Pareto frontier. As can be seen from the figure, in flat regions of the Pareto frontier only a few EP-optimal points have been generated, whereas many points have been produced in highly curved regions.

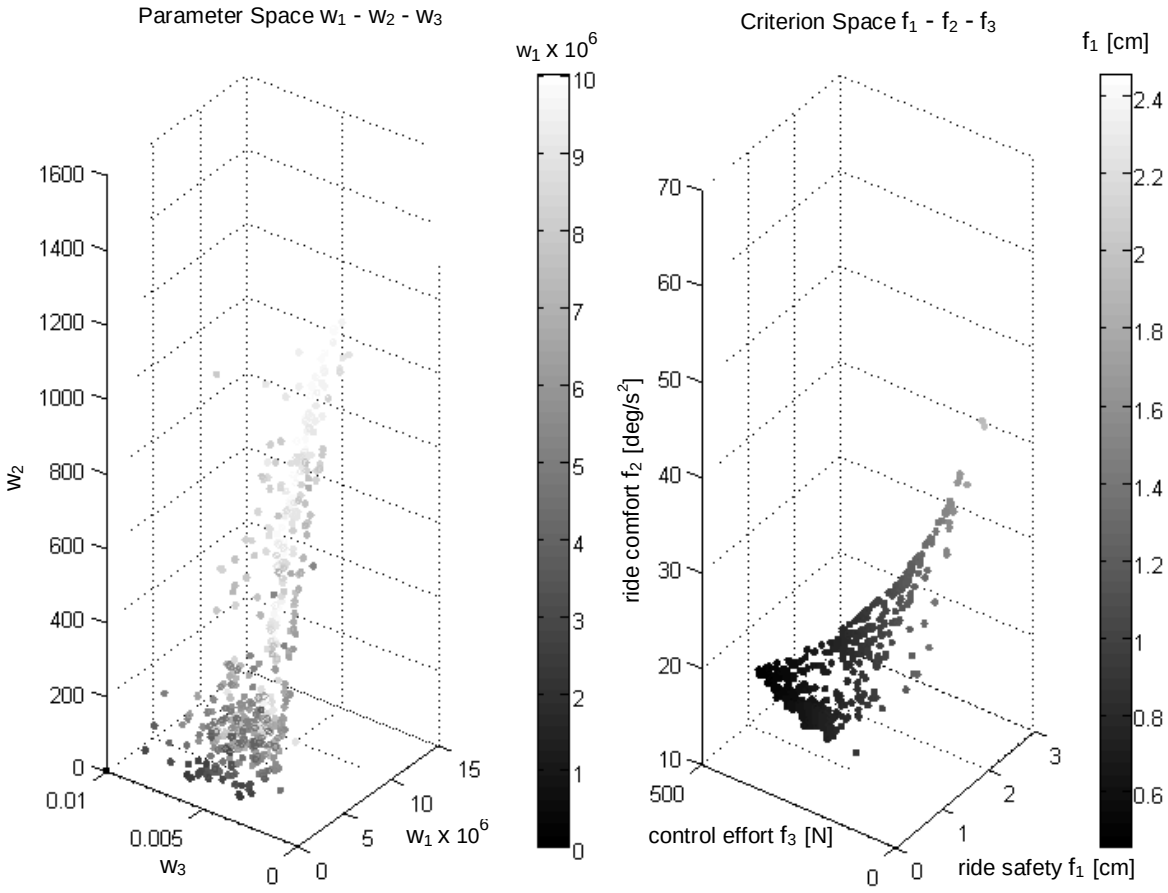

Figure 5.10: Optimization results based on recursive knee approach

Table 5.3: EP-optimal solutions for active suspension optimization based on recursive knee approach

point	criteria			design variables		
	f_1 $[cm]$	f_2 $[deg/s^2]$	f_3 $[N]$	w_1 x10^4	w_2	w_3 x10^{-4}
1	0.4541	17.6671	426.5835	6.9832	20.4031	0.3543
2	0.4586	16.8939	423.7608	5.6952	70.6110	0.2881
3	0.4590	17.0350	423.7857	5.7814	22.5580	0.2935
4	0.4593	16.5736	422.5167	5.6077	12.1690	0.2813
5	0.4597	16.7113	422.5151	5.7207	13.3746	0.3234
6	0.4685	16.7059	415.5217	5.9585	10.5000	0.3371
7	0.4733	17.1598	412.7485	5.9763	13.0354	0.3432
8	0.4814	17.4234	407.0224	5.5970	18.2439	0.3275
9	0.4817	16.0460	403.9879	4.5710	48.9117	0.2274
10	0.4845	16.5221	402.7689	6.2455	42.8665	0.3173
…	…	…	…	…	…	…
320	0.7857	30.5169	55.5835	2.7803	17.8988	0.6568
321	0.7918	30.2930	53.5705	2.9684	21.4410	0.7175
322	0.8038	29.7068	49.7925	3.6794	67.9262	0.9308
323	0.8049	29.5808	49.4576	2.5385	51.9351	0.6450
324	0.8292	31.8725	43.4565	2.9824	17.8921	0.8113
325	0.8418	32.2900	40.2773	2.3654	15.7321	0.6718
326	0.8448	32.5263	39.5997	2.3689	12.0562	0.6786
327	0.8869	31.4941	29.7252	2.9643	97.0776	0.5020
328	0.9819	36.3711	9.6581	4.2049	92.5754	0.6611
329	1.0014	35.8023	7.7729	4.9348	88.5600	0.7065
330	1.2586	48.1030	0.0000	0.0000	0.00000	0.0100

Since the knee of the Pareto frontier is usually located in the middle interval, the formula (5.16) for defining starting points used in the compromise method is inefficient to the recursive knee approach. A new formulation for defining the starting point of the knee sub-problem (5.13) has been presented by Wachal and Bestle [86]. With this formulation, the starting point can be changed automatically when the sub-problem (5.13) is required to be restarted after a definite number of non-terminal computations. Another advantage of the recursive knee approach is the Pareto filter algorithm which filters out non-EP-optimal solutions as well as local EP-optimal solutions that may have been obtained during the generation of candidate designs.

Unlike the compromise method, the recursive knee approach requires to solve the single optimization problems (5.8) only once for the individual minima defining the first CHIM. Moreover, the risk of obtaining non-EP-optimal solutions and local EP-optimal solutions is avoided by the Pareto filter algorithm. The recursive knee approach, therefore, reduces the computational time by about three times compared to the compromise method while giving the same results.

5.5 MCO Problem for Passive and Active Suspensions

In the previous section, the active suspension has been optimized with fixed passive suspension parameters. In order to estimate the effectiveness of optimizing the passive and active suspensions at the same time, the three-criterion optimization problem where both passive and active suspension parameters are considered as design variables is introduced in this section.

5.5.1 Problem definition

With inclusion of LQR algorithm introduced in section 5.4.2, the optimal passive and active suspension parameters for the spatial car model can be obtained by solving the following MCO problem:

$$\min_{\boldsymbol{p}\in\mathcal{P}} \boldsymbol{f}(\boldsymbol{p}), \quad \boldsymbol{f}(\boldsymbol{p}) = [\, f_1,\ f_2,\ f_3\,]^{\boldsymbol{T}},$$

$$\mathcal{P} := \left\{ \boldsymbol{p} = \left[\, \boldsymbol{p}_1^{\boldsymbol{T}},\ \boldsymbol{p}_2^{\boldsymbol{T}} \,\right]^{\boldsymbol{T}} \in I\!R^{9} \;\middle|\; \boldsymbol{p}^{\boldsymbol{l}} \le \boldsymbol{p} \le \boldsymbol{p}^{\boldsymbol{u}},\ h(\boldsymbol{p}) \le 0 \right\}$$

where

$$\boldsymbol{p}_1 = \left[\, b_{\boldsymbol{f}},\ b_{\boldsymbol{r}},\ k_{\boldsymbol{f}},\ k_{\boldsymbol{r}},\ r_{\boldsymbol{f}},\ r_{\boldsymbol{r}} \,\right]^{\boldsymbol{T}} \in I\!R^{6},$$

$$\boldsymbol{p}_2 = [\, w_1,\ w_2,\ w_3\,]^{\boldsymbol{T}} \in I\!R^{3},$$

$$\boldsymbol{p}_1^{\boldsymbol{l}} = [1000,\ 1000,\ 15000,\ 15000,\ 15000,\ 5000]^{\boldsymbol{T}},$$

$$\boldsymbol{p}_1^{\boldsymbol{u}} = [\, 2000,\ 2000,\ 25000,\ 25000,\ 25000,\ 15000]^{\boldsymbol{T}},$$

$$\boldsymbol{p}_2^{\boldsymbol{l}} = \left[\, 0,\ 0,\ 10^{-6} \,\right]^{\boldsymbol{T}},$$

$$\boldsymbol{p}_2^{\boldsymbol{u}} = \left[10^{8},\ 10^{8},\ 10^{2} \right]^{\boldsymbol{T}},$$

$$h(\boldsymbol{p}) := \max_{\boldsymbol{i}} \left(\max_{\boldsymbol{t}} \left| u_{\boldsymbol{i}}(t) \right| \right) - 500. \qquad (5.22)$$

In the above formulation, the vector of design variables $\boldsymbol{p}_1$ includes the passive suspension parameters while $\boldsymbol{p}_2$ contains weighting factors which will determine the optimal control gains $\boldsymbol{K}_{\boldsymbol{x}}^{*}$ and $\boldsymbol{K}_{\boldsymbol{w}}^{*}$ for the active suspension based on the LQR algorithm.

5.5.2 Optimization results

In order to derive comparable results with a reduced computational time, the compromise method for the case of arbitrary control effort is applied to solve the problem (5.22), which is concretized by the MATLAB.m files introduced in Appendix D.1. The optimization results are represented by the asterisks in Figure 5.11 and given in Table 5.4. Compared to the results obtained from optimizing the passive suspension denoted by the stars in the figure, ride safety and ride comfort of the car are improved by about 60% and 75%, respectively. These values are 14% and 2% better than the results derived from optimizing the active suspension only with arbitrary control effort which are described by the six-pointed stars.

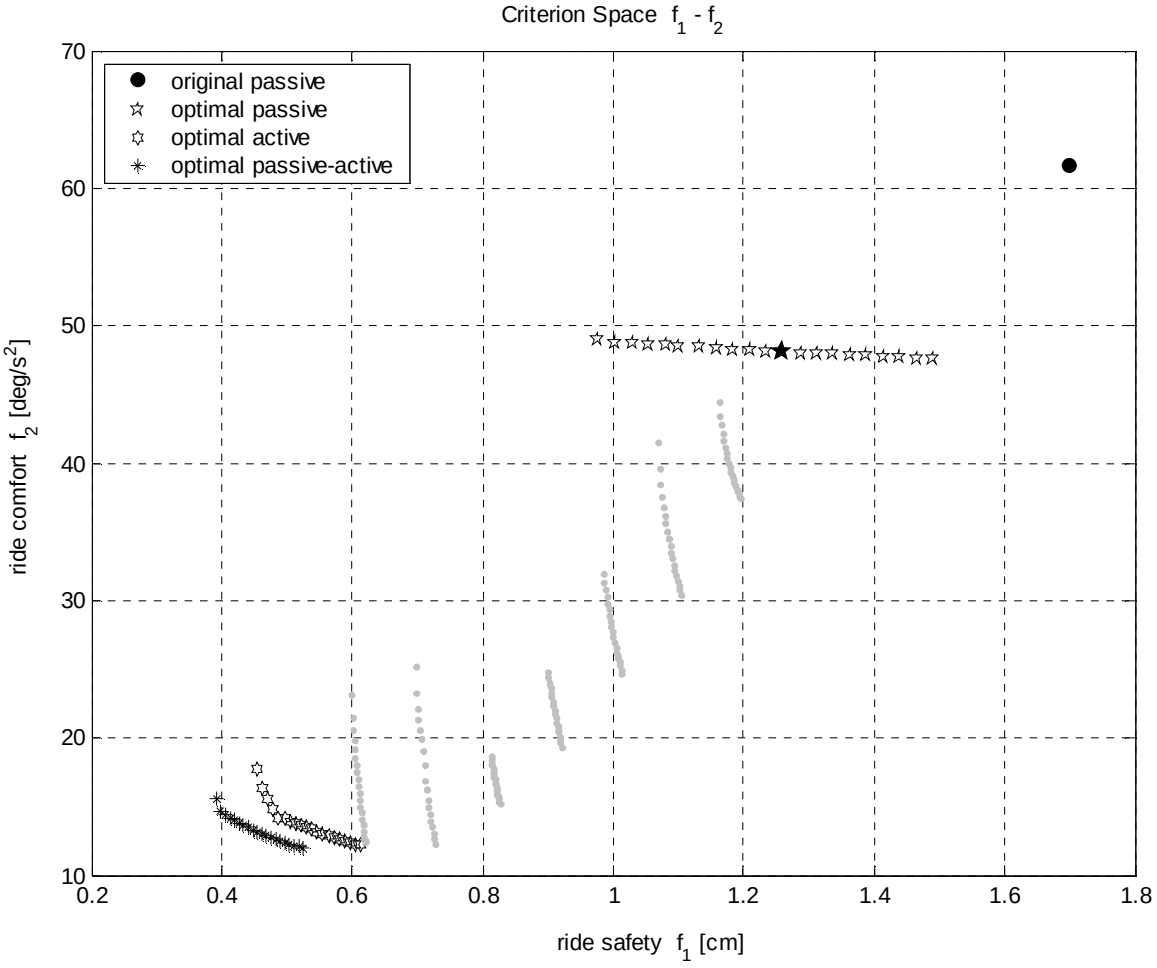

Figure 5.11: Comparison between the optimized passive, optimized active and optimized passive-active suspensions

Table 5.4: EP-optimal solutions for passive-active suspension optimization based on compromise method for arbitrary control effort

point	criteria			design variables		
	$f_1\,[cm]$	$f_2\,[deg/s^2]$	$f_3\,[N]$	w_1 x10^4	w_2	w_3 x10^{-4}
1	0.5243	11.9999	334.0244	4.6723	24.6195	0.3730
2	0.5173	12.1472	337.9501	4.6723	20.7456	0.3219
3	0.5104	12.3599	341.8873	4.6723	17.2809	0.2770
4	0.4928	12.4905	351.9964	4.6723	13.2872	0.2195
5	0.4931	12.4921	351.8148	4.6723	13.3267	0.2202
6	0.4894	12.5059	354.0309	4.6723	12.6984	0.2108
7	0.4825	12.6178	358.0819	4.6723	11.3196	0.1919
8	0.4755	12.7886	362.0795	4.6723	9.9694	0.1740
9	0.4685	12.9479	366.1000	4.6723	8.8603	0.1590
10	0.4615	13.1058	370.1430	4.6723	7.9116	0.1461
11	0.4546	13.1562	374.1754	4.6723	7.0495	0.1342
12	0.4476	13.3162	378.2476	4.6723	6.3341	0.1244
13	0.4406	13.4795	382.3260	4.6723	5.7025	0.1156
14	0.4336	13.6466	386.4136	4.6723	5.1411	0.1078
15	0.4267	13.8180	390.5094	4.6723	4.6399	0.1008
16	0.4197	13.9950	394.6172	4.6723	4.1910	0.0946
17	0.4127	14.1766	398.7333	4.6723	3.7871	0.0890
18	0.4057	14.3632	402.8607	4.6723	3.4239	0.0840
19	0.3988	14.6071	406.9448	4.6723	3.0608	0.0794
20	0.3918	15.5962	410.3890	4.6723	2.3471	0.0736

point	design variables					
	b_f	b_r	k_f	k_r	r_f	r_r
	x10^3 $[Ns/m]$		x10^4 $[N/m]$		x10^4 $[N/m]$	
1	2.0000	2.0000	2.4979	2.4943	2.5000	1.5000
2	2.0000	2.0000	2.4982	2.4945	2.5000	1.5000
3	2.0000	2.0000	2.4982	2.4945	2.5000	1.5000
4	1.9999	1.9999	2.4988	2.4946	2.5000	1.5000
5	1.9993	1.9992	2.4992	2.4949	2.5000	1.5000
6	2.0000	2.0000	2.4982	2.4945	2.5000	1.5000
7	2.0000	2.0000	2.4982	2.4945	2.5000	1.5000
8	2.0000	2.0000	2.4982	2.4945	2.5000	1.5000
9	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
10	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
11	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
12	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
13	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
14	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
15	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
16	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
17	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
18	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
19	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000
20	2.0000	2.0000	2.4983	2.4945	2.5000	1.5000

The obtained results show that the best solution for designing the active suspension system is to optimize the parameters of both passive and active components at the same time. However, it should be noted that the derived optimal values are valid only for a specific operation point defined by given values of the car yaw rate and lateral acceleration. For different operation points, the passive and active suspension parameters have to change appropriately. Since the passive suspension parameters are typically unchangeable, the practical way is to design a controller that is able to change its parameters according to the changes of the system's varying parameters. The process of defining such a controller is known as gain-scheduling control design which will be introduced in the next chapter in more details.

Chapter 6

Gain-scheduling Control

Gain-scheduling is one of the most common control techniques used to design controllers that can change their parameters in response to the variation of the system dynamics resulting from varying parameters. In this chapter, gain-scheduling control will be designed for the spatial car model. By considering the effects of suspension and tire deformation on the vehicle stability in cornering situations, the *operation region* of the spatial car model will be determined. Based on *optimal local controllers* defined for *selected operation points*, the *parameter-dependent controller* will be then designed. The effectiveness of the designed parameter-dependent controller will be demonstrated through the simulation of *double-lane-change maneuvers*, which are designated as standard vehicle handling tests. In order to find optimal paths for the double-lane-change maneuvers that minimize the vehicle lateral dynamics during the test, a *path generation problem* will be formulated and solved in this chapter.

6.1 Operation Region of the Spatial Car Model

As indicated in Chapter 3, the linear parameter-varying (LPV) spatial car model (3.76) can be simplified as a set of linear time-invariant (LTI) systems (3.78) defined by pairs $(\dot{\gamma}, a_y)$. In this section, upper bounds for the lateral acceleration a_y and yaw rate $\dot{\gamma}$, which determine the operation region of the spatial car model, will be defined based on the stability condition of vehicles in cornering, taking into account the compliance of suspension and tires.

6.1.1 Vehicle side-slip and rollover stability

When cornering with large lateral acceleration, the car may *slip sideways* or *roll over* under the action of the centrifugal force located at the car center of gravity. The most traditional analysis of vehicle side-slip and rollover stability is based on a rigid vehicle model in steady-state cornering as shown in Figure 6.1.

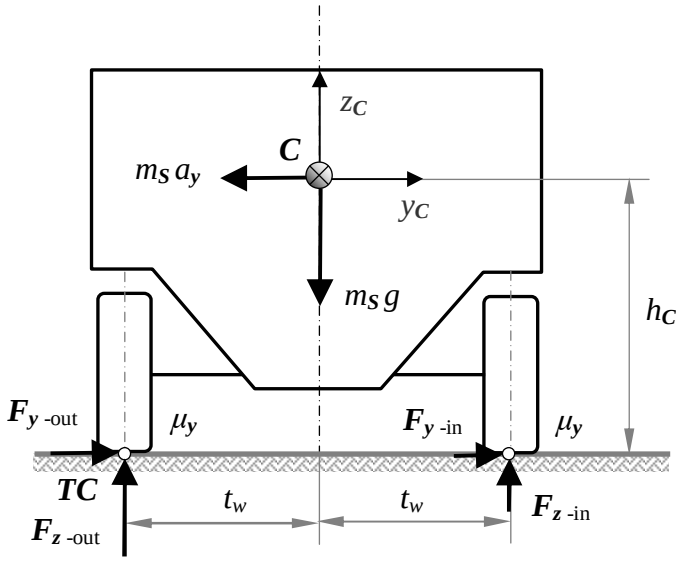

Figure 6.1: Rigid vehicle model

During cornering, the centrifugal force $m_S\, a_y$ is counterbalanced by the lateral tire forces, which are limited by values proportional to the corresponding normal tire loads and the tire-ground friction coefficient, Mitschke [51] and Steinberg [77]. It is assumed that the road friction coefficients μ_y are the same for all four wheels, resulting in

$$m_S\ a_y = F_{y\text{-out}} + F_{y\text{-in}} \le \mu_y \left(F_{z\text{-out}} + F_{z\text{-in}}\right) = \mu_y\, m_S\ g$$

or

$$a_{y\,slip} \le \mu_y\, g \tag{6.1}$$

where g = 9.81 m/s^2 is the gravitational acceleration. For dry-asphalt roads, the friction coefficient is approximately μ_y = 1.0 resulting in a maximum lateral acceleration value $a_y \le a_{y\,slip}$ = 9.81 m/s^2 that the car is able to reach before slipping sideways.

The vehicle rollover threshold is defined from the two-wheel lift condition, i.e. the normal loads on the inside wheels reach zeros, and thus the total normal load on the inside wheels is $F_{z\text{-in}} = 0$. Taking moments about the center of contact patches for the outside wheels *TC* with the simplification of the same front and rear half-track width, $t_{wf} = t_{wr} = t_w$, results in the simple formula for the lateral acceleration at the rollover threshold:

$$a_y \le a_{y\,roll} = \frac{t_w}{h_C} g \tag{6.2}$$

where h_C is the static height of car's CG above ground. The ratio (t_w/h_C) is often referred to as the static stability factor used to estimate roughly the vehicle rollover-resistance ability.

The maximum lateral acceleration a_y^u of the car during cornering is limited by the smaller one of the lateral accelerations at the rollover and side-slip threshold, i.e.

$$a_y^u = \min\left(a_{y\,roll},\ a_{y\,slip}\right). \tag{6.3}$$

For the spatial car model with the dates given in Appendix A, $a_{y\,roll} = 14.91\ m/s^2$ is greater than $a_{y\,slip} = 9.81\ m/s^2$, which means that the car will more probably slip sideways instead of rolling over.

However, neglecting the deformation of suspension and tires in the above analysis often leads to overestimation of the rollover threshold. To determine more accurately the upper bound for the lateral acceleration, the influences of suspension and tire compliance on the vehicle rollover stability must be considered.

6.1.2 Effects of suspension and tires compliance

Due to lateral compliance of suspension and tires, the distance in lateral direction between the center-plane of the vehicle and the tire contact patches is changed, usually reduced. In addition, the transmission of lateral forces between the body and the wheels results in vertical components called "jacking" forces, which in general do not cancel out and usually increase the static height of car's CG. The changes in effective half-track width and height of car's CG, which reduce the lateral acceleration at the rollover threshold, are illustrated in Figure 6.2.

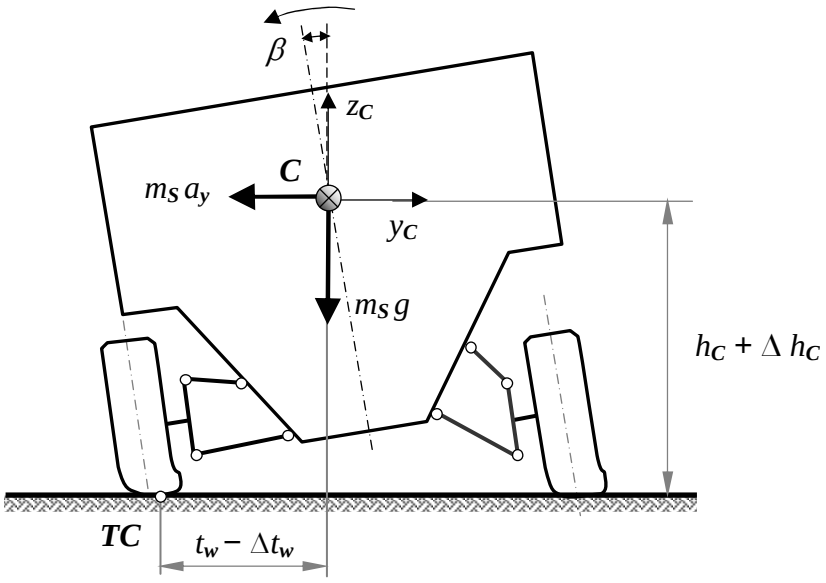

Figure 6.2: Rollover model with deformable suspension and tires

The changes in effective half-track width result from body roll, tire deformation and suspension kinematics. During cornering, the car body rolls about the roll axis resulting in a lateral shift of the car's CG towards outside of turn which decreases the effective half-track. Also, the half-track width is reduced due to the lateral displacements of the tire contact patches with respect to the body resulting from lateral distortion of tires. An additional change of half-track width occurs because of suspension kinematics. During suspension deflection, the wheel changes its inclination angle with respect to the car body, resulting in a lateral displacement of the wheel which increases the effective half-track width.

During cornering maneuvers on smooth roads, the vehicle body is usually subjected to vertical forces, often referred to as "jacking" forces, which tend to lift the car's CG above the static location. In steady-state cornering there are primarily two sources of jacking forces: nonlinearities in suspension stiffness characteristics and vertical components of forces transmitted by suspension links. Suspension stiffness characteristics are usually progressive, that is stiffness increases with suspension deflection in order to maintain good ride properties with full load. This characteristic of suspension permits smaller deflection in compression of the outside suspension than deflection in extension of the inside suspension. As a result, the height of car's CG increases. This effect is highly dependent on the

particular stiffness characteristic, which is difficult to be captured in a general approach. Therefore, it is often neglected in the analysis of influences of suspension compliance on the vehicle rollover stability. The second jacking effect is a result of forces in suspension links. Lateral forces generated during cornering maneuvers are transmitted between the body and the wheels through relatively rigid suspension links, which are in general not parallel to the ground. Therefore, the reaction forces in these elements have vertical components resulting in a vertical net force, which usually pushes the body up for typical suspensions.

Hac [35] shows that the effect of total change in the effective half-track width resulting from body roll, tire compliance and suspension kinematics reduces the lateral acceleration at the rollover threshold computed by the static stability factor (6.2) by about 15%, while the change in height of car's CG caused by jacking forces contributes a reduction of about 5% for passenger cars. Applying these results to the spatial car model yields the lateral acceleration at the rollover threshold $a_{y\ roll} \approx 11.93\ m/s^2$. This result is limited to the passive suspension system. When the active suspension system is applied, the suspension deformation will be restrained significantly by the controlled forces generated by the active dampers at four wheels. In this case, the value of $a_{y\ roll}$ will be between 11.93 −14.91 m/s^2.

Since the lateral acceleration at the rollover threshold is greater than that at the side-slip threshold, the upper bound value of the lateral acceleration for the spatial car model is determined as $a_y^{\boldsymbol{u}} = a_{y\ slip} = 9.81\ m/s^2$.

6.1.3 Definition of operation region and operation points

Theoretically, the *operation region* of the spatial car model is a rectangle bounded by the coordinate axes and the upper bounds for lateral acceleration $a_y^{\boldsymbol{u}}$ and yaw rate $\dot{\gamma}^{\boldsymbol{u}}$ as illustrated in Figure 6.3, where $a_y^{\boldsymbol{u}} = 9.81\ m/s^2$ is the maximum lateral acceleration that the car is able to reach before slipping sideways, while $\dot{\gamma}^{\boldsymbol{u}}$ can be computed from $a_y^{\boldsymbol{u}}$ and the car minimum turning radius $R_{\min}$ as

$$\dot{\gamma}^{\boldsymbol{u}} = \sqrt{\frac{a_y^{\boldsymbol{u}}}{R_{\min}}}\ . \tag{6.4}$$

This equation is obtained from (3.51) in the limit case $a_y^u = v\,\dot{\gamma}^u$ and $v = R_{\min}\,\dot{\gamma}^u$. With $R_{\min}$ = 10 *m*, the upper bound of yaw rate is derived as $\dot{\gamma}^u \approx 1.0$ *rad/s*. In practice, this operation region may be further reduced by minimum turning radius and maximum velocity as $a_y \le v_{\max}\,\dot{\gamma}$, however, in order to simplify the computational procedure, the whole operation region as shown in Figure 6.3 will be used to design gain-scheduling control for the spatial car model.

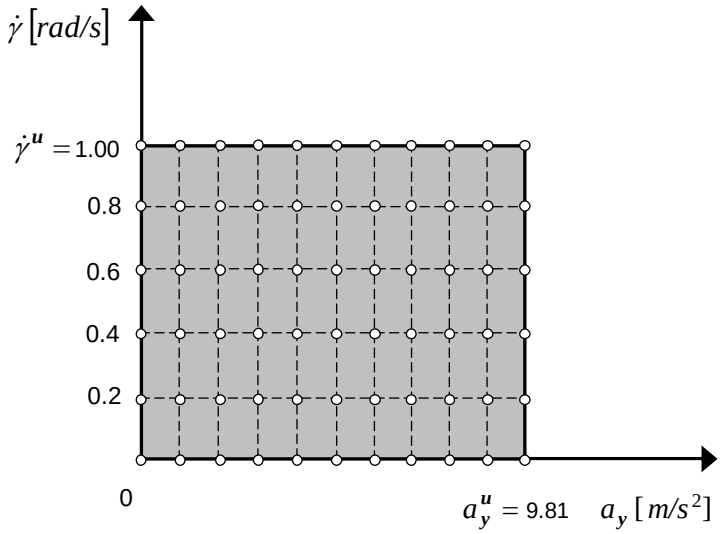

Figure 6.3: Operation region and selected operation points

The variation of the car yaw rate $\dot{\gamma}$ and lateral acceleration a_y in the operation region leads to changes in the system dynamics. To maintain the required performance of the system, the controllers have to update their parameters on-line according to varying system parameters. Thus, a parameter-dependent state-feedback gain matrix $\boldsymbol{K}_x(\dot{\gamma}, a_y)$ and disturbance-feed forward gain vector $\boldsymbol{K}_w(\dot{\gamma}, a_y)$ are required to control the spatial car model. To design such a controller, the dependence of the controller on the variation of $\dot{\gamma}$ and a_y over the operation region must be investigated. For this purpose, 66 pairs $(\dot{\gamma}, a_y)$ corresponding to 66 *operation points* are selected as illustrated by the small circles in Figure 6.3. Each operation point defines a linear time-invariant system, whose optimal controllers can be obtained by applying the multi-criterion optimization method with the linear quadratic regulator algorithm as introduced in the previous chapter.

6.2 Gain-scheduling Design for the Spatial Car Model

In this section, the *optimal local controllers*, i.e. the optimal state-feedback gain matrices and disturbance-feed forward gain vectors of the linear time-invariant systems defined by the selected operation points, will be computed. The negligible influence of the yaw rate on the optimal local controllers will be shown. Based on the variation laws of the components of the optimal local controllers, the *parameter-dependent controller* as continuous functions of the lateral acceleration will be formulated.

6.2.1 Optimal local controllers

In order to simplify the problem of finding the optimal local controllers, a scalar optimization problem is used instead of the multi-criterion optimization problem. Since ride safety is most important for cars in cornering, this criterion is defined as the only objective for the gain-scheduling design:

$$\min_{\boldsymbol{p} \in \mathcal{P}} \; f_1 \quad \textit{where} \quad f_1 = \sqrt{\frac{1}{4T} \int_0^{\boldsymbol{T}} \sum_{\boldsymbol{i}=1}^{4} z_{\boldsymbol{Si}}^2(t)\, dt} \;, \qquad i = 1(1)4 \tag{6.5}$$

where the vector of design variables $\boldsymbol{p}$ and the feasible design space $\mathcal{P}$ are defined according to (5.18). The optimal state-feedback gain matrix $\boldsymbol{K}_x$ and disturbance-feed forward gain vector $\boldsymbol{K}_w$ are derived from the optimal solution of problem (6.5) based on a strategy similar to the one shown in Figure 5.8.

Figure 6.4 shows the obtained results for the selected operation points, which are generated by changing evenly lateral acceleration a_y while keeping constant yaw rate $\dot{\gamma}$. The small figures describe the dependences of the optimal control gains $\boldsymbol{K}_x$ and $\boldsymbol{K}_w$ on lateral acceleration and yaw rate. As can be seen from the figures, the obtained optimal local controllers depend only slightly on the yaw rate. This can be explained by the negligible effect of yaw rate on the car ride safety criterion as follows:

> With values $\dot{\gamma} \le 1.0\, rad/s$ for the yaw rate and the parameters of the car given in Appendix A.1, the yaw rate-dependent matrices $\boldsymbol{P}(\dot{\gamma})$ in (3.56) and $\boldsymbol{Q}(\dot{\gamma})$ in (3.57) (with $a_x = 0$ and $\ddot{\gamma} = 0$) are much smaller than the invariant matrices $\boldsymbol{G}_S \boldsymbol{B}_S \boldsymbol{G}_S^T$ and $\boldsymbol{G}_S (\boldsymbol{K}_S + \boldsymbol{K}_A) \boldsymbol{G}_S^T$ in the state equations (3.74), respectively. The matrices $\boldsymbol{P}(\dot{\gamma})$ and

$\boldsymbol{Q}(\dot{\gamma})$, therefore, can be omitted in the matrices $\boldsymbol{A}(\dot{\gamma})$ and $\boldsymbol{C}(\dot{\gamma})$ defined in equations (3.74) and (3.75), i.e.

$$\boldsymbol{A}(\dot{\gamma}) \approx \boldsymbol{A}_0 := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{G_S}\left(\boldsymbol{K_S}+\boldsymbol{K_A}\right)\boldsymbol{G_S^T} & -\boldsymbol{M}^{-1}\boldsymbol{G_S}\boldsymbol{B_S}\boldsymbol{G_S^T} \end{bmatrix} \text{ and}$$

$$\boldsymbol{C}(\dot{\gamma}) \approx \boldsymbol{C}_0 := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{g}_{\ddot{\beta}}^{\boldsymbol{T}}\,\boldsymbol{M}^{-1}\boldsymbol{G_S}\left(\boldsymbol{K_S}+\boldsymbol{K_A}\right)\boldsymbol{G_S^T} & -\boldsymbol{g}_{\ddot{\beta}}^{\boldsymbol{T}}\,\boldsymbol{M}^{-1}\boldsymbol{G_S}\boldsymbol{B_S}\boldsymbol{G_S^T} \end{bmatrix}. \tag{6.6}$$

Thus, the linear system (3.76) can be substituted by

$$\begin{cases} \dot{\boldsymbol{x}} \approx \boldsymbol{A}_0\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} + \boldsymbol{b_w}\,w(a_y), \\ \\ \boldsymbol{y} \approx \boldsymbol{C}_0\,\boldsymbol{x} + \boldsymbol{D}\,\boldsymbol{u} + \boldsymbol{d_w}\,w(a_y). \end{cases} \tag{6.7}$$

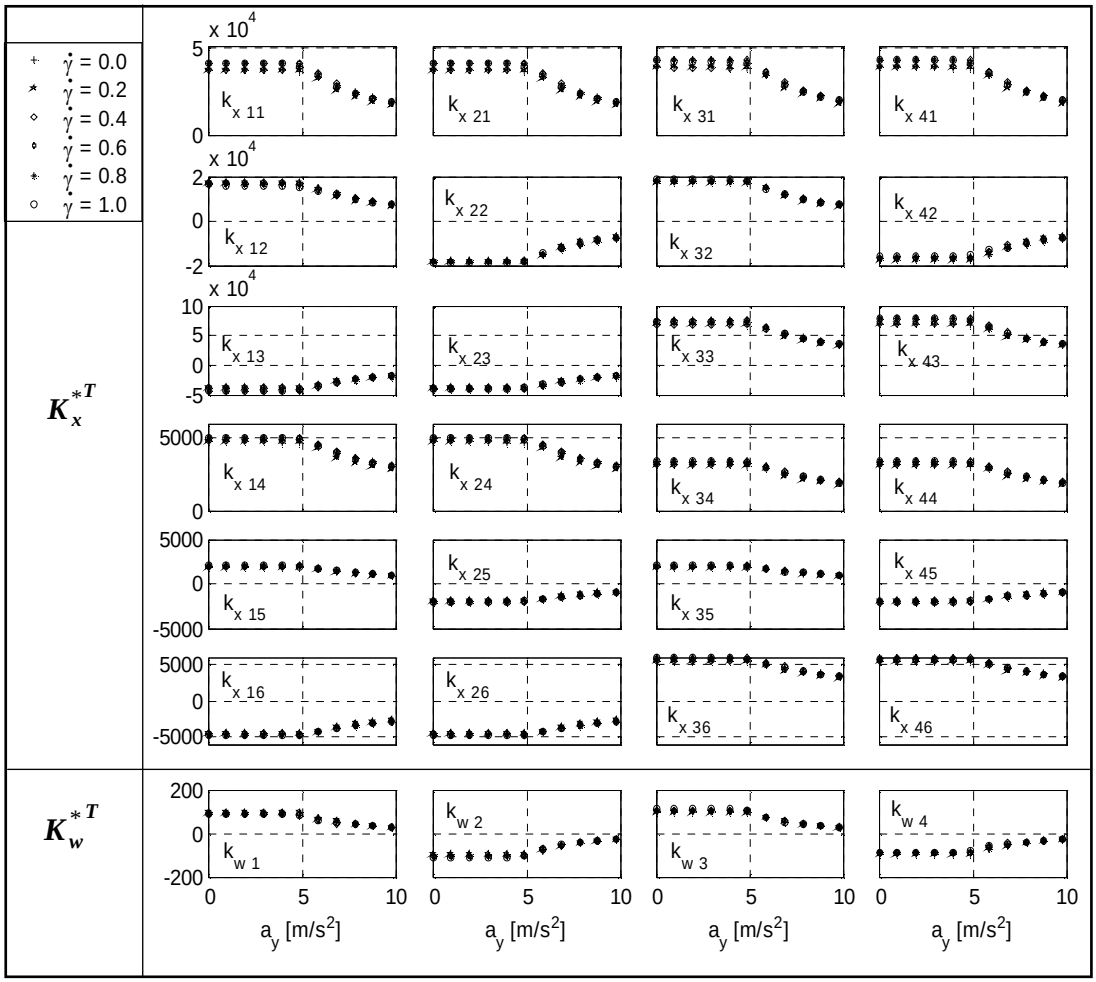

Figure 6.4: Components of optimal gains $\boldsymbol{K}_{\boldsymbol{x}}^{*}$ and $\boldsymbol{K}_{\boldsymbol{w}}^{*}$ vs. a_y for different $\dot{\gamma}$

Since then the state vector $\boldsymbol{x}$ does not depend on the yaw rate $\dot{\gamma}$, the vector of measured outputs $\boldsymbol{y} = [\boldsymbol{z_S^T},\ \ddot{\beta}]^{\boldsymbol{T}}$, and thus the car ride safety criterion f_1 is independent of $\dot{\gamma}$.■

The validity of this approximation can be seen in Figure 6.7, from the negligible effects of yaw rate on the suspension displacement z_{S1} of the front-left wheel. The same results are also observed for the suspension displacements of front-right, rear-left and rear-right wheels. Since the car ride safety criterion f_1 is a function of the suspension displacements z_{Si}, its value is almost constant when varying the yaw rate. The optimal solutions of the optimization problem (6.5), therefore, do not depend on the change of yaw rate as seen in Figure 6.4.

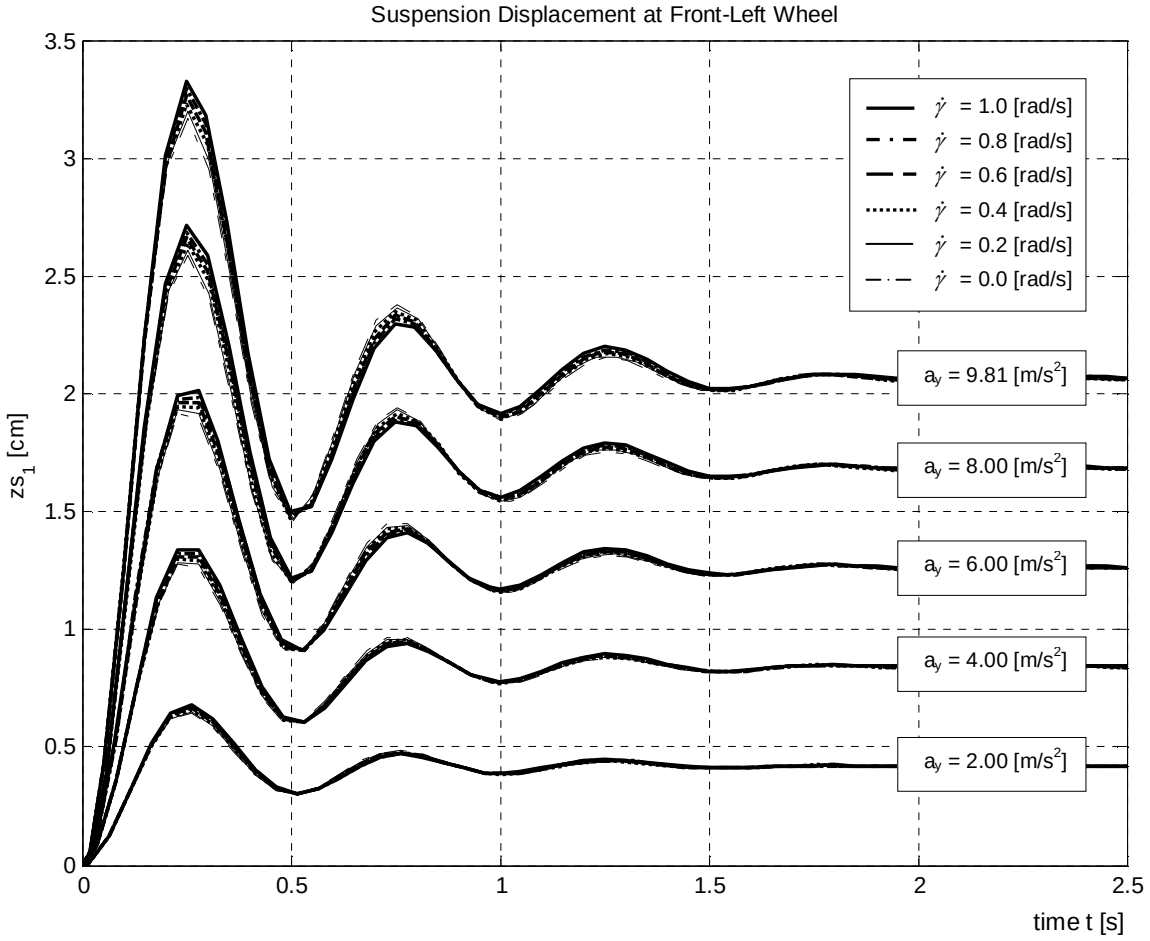

Figure 6.5: Negligible effects of yaw rate $\dot{\gamma}$ on suspension displacement at front-left wheel

6.2.2 Parameter-dependent controller design

As discussed above, the effect of the yaw rate on the optimal local controllers can be neglected. However, to design an adequate mean controller, the mean values of the optimal local controllers obtained from six different values of yaw rate are used for the parameter-dependent controller of the spatial car model. The values of the control gains are given in Table 6.1 and shown by the dots in Figure 6.6. In order to obtain continuously varying functions of the lateral acceleration, polynomials in a_y fitting the mean values of the optimal local controllers have to be found.

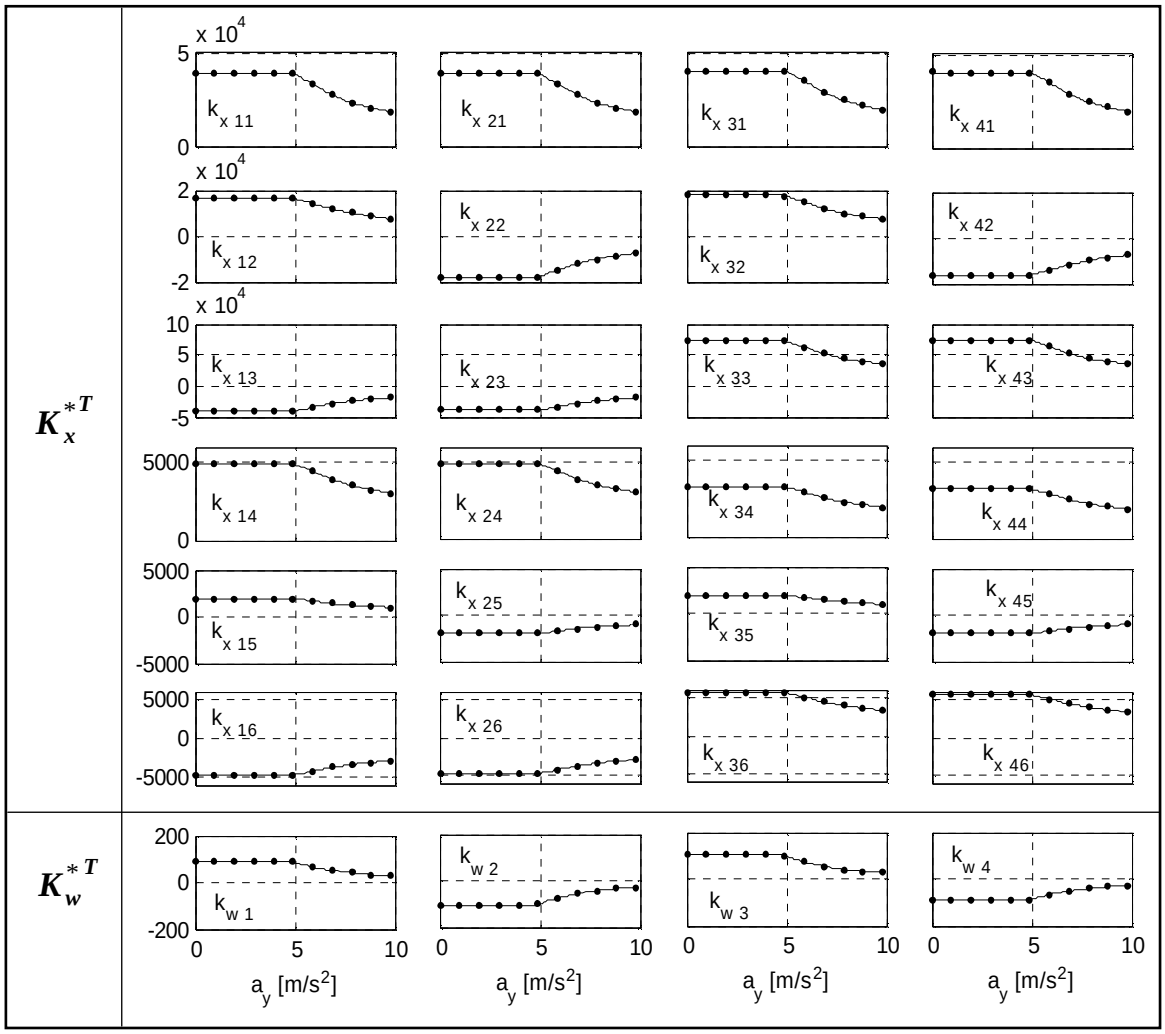

Figure 6.6: Mean-value points and interpolated curves for control gains

Table 6.1: Mean values of the optimal local controllers (x10^4)

	lateral acceleration a_y $[m/s^2]$										
	0.0000	0.9810	1.9620	2.9430	4.9050	4.9050	5.8860	6.8670	7.8480	8.8290	9.8100
$k_{x\,11}$	3.8874	3.8833	3.8833	3.8833	3.8813	3.8822	3.3479	2.7419	2.3430	2.0516	1.8212
$k_{x\,12}$	1.6879	1.6858	1.6859	1.6858	1.6846	1.6516	1.4313	1.1867	1.0060	0.8560	0.7414
$k_{x\,13}$	-4.0186	-4.0143	-4.0143	-4.0143	-4.0123	-4.0111	-3.4292	-2.7747	-2.3615	-2.0672	-1.8344
$k_{x\,14}$	0.4892	0.4889	0.4889	0.4889	0.4887	0.4888	0.4462	0.3913	0.3544	0.3257	0.3016
$k_{x\,15}$	0.1943	0.1941	0.1941	0.1941	0.1941	0.1946	0.1714	0.1429	0.1238	0.1099	0.0987
$k_{x\,16}$	-0.4668	-0.4665	-0.4665	-0.4665	-0.4663	-0.4663	-0.4259	-0.3734	-0.3380	-0.3104	-0.2872
$k_{x\,21}$	3.8959	3.8917	3.8917	3.8917	3.8897	3.8904	3.3522	2.7427	2.3430	2.0516	1.8212
$k_{x\,22}$	-1.8095	-1.8073	-1.8074	-1.8073	-1.8062	-1.7716	-1.4987	-1.2005	-1.0060	-0.8560	-0.7414
$k_{x\,23}$	-3.8193	-3.8152	-3.8153	-3.8152	-3.8132	-3.8162	-3.3227	-2.7539	-2.3615	-2.0672	-1.8344
$k_{x\,24}$	0.4887	0.4884	0.4884	0.4884	0.4882	0.4883	0.4459	0.3912	0.3544	0.3257	0.3016
$k_{x\,25}$	-0.1934	-0.1932	-0.1932	-0.1932	-0.1931	-0.1937	-0.1709	-0.1428	-0.1238	-0.1099	-0.0987
$k_{x\,26}$	-0.4674	-0.4671	-0.4671	-0.4671	-0.4669	-0.4672	-0.4261	-0.3734	-0.3380	-0.3104	-0.2872
$k_{x\,31}$	4.0485	4.0443	4.0443	4.0443	4.0423	4.0433	3.5008	2.8820	2.4743	2.1754	1.9381
$k_{x\,32}$	1.8045	1.8024	1.8024	1.8024	1.8012	1.7667	1.4920	1.1920	0.9980	0.8492	0.7356
$k_{x\,33}$	7.2091	7.2016	7.2017	7.2016	7.1980	7.2016	6.2683	5.1940	4.4687	3.9294	3.5014
$k_{x\,34}$	0.3254	0.3252	0.3252	0.3252	0.3251	0.3252	0.2948	0.2563	0.2304	0.2103	0.1936
$k_{x\,35}$	0.1919	0.1917	0.1917	0.1917	0.1916	0.1922	0.1696	0.1417	0.1228	0.1090	0.0979
$k_{x\,36}$	0.5604	0.5600	0.5600	0.5600	0.5598	0.5601	0.5076	0.4411	0.3961	0.3613	0.3322
$k_{x\,41}$	4.0569	4.0527	4.0527	4.0527	4.0507	4.0514	3.5051	2.8828	2.4743	2.1754	1.9381
$k_{x\,42}$	-1.6653	-1.6633	-1.6633	-1.6632	-1.6621	-1.6295	-1.4149	-1.1763	-0.9980	-0.8492	-0.7356
$k_{x\,43}$	7.4068	7.3991	7.3991	7.3991	7.3955	7.3950	6.3740	5.2147	4.4687	3.9294	3.5014
$k_{x\,44}$	0.3249	0.3247	0.3247	0.3247	0.3246	0.3247	0.2945	0.2563	0.2304	0.2103	0.1936
$k_{x\,45}$	-0.1927	-0.1926	-0.1926	-0.1926	-0.1925	-0.1930	-0.1700	-0.1418	-0.1228	-0.1090	-0.0979
$k_{x\,46}$	0.5598	0.5594	0.5594	0.5594	0.5592	0.5592	0.5073	0.4410	0.3961	0.3613	0.3322
k_{w1}	0.0091	0.0091	0.0091	0.0091	0.0091	0.0087	0.0067	0.0053	0.0042	0.0033	0.0028
k_{w2}	-0.0104	-0.0103	-0.0103	-0.0103	-0.0103	-0.0100	-0.0075	-0.0055	-0.0042	-0.0033	-0.0028
k_{w3}	0.0102	0.0102	0.0102	0.0102	0.0102	0.0099	0.0074	0.0055	0.0041	0.0033	0.0027
k_{w4}	-0.0091	-0.0090	-0.0090	-0.0090	-0.0090	-0.0087	-0.0067	-0.0053	-0.0041	-0.0033	-0.0027

This is achieved with the MATLAB function "*polyfit*", which finds the coefficients of a polynomial $p(x)$ of order n in a least-mean-square sense. To reduce the order of the interpolation functions, the fitting curves illustrated in Figure 6.6 are divided into two parts, respectively: 1) curve segments corresponding to $a_y > 4.905\ m/s^2$ which are represented by fourth-order polynomial functions of the form $p(a_y) = c_1 a_y^4 + c_2 a_y^3 + c_3 a_y^2 + c_4 a_y + c_5$, and 2) straight-line segments for $a_y \leq 4.905\ m/s^2$ with magnitudes defined from the associated polynomial functions for $p(a_y = 4.905)$. The coefficients of the fourth-order polynomials functions are given in Table 6.2.

Table 6.2: Interpolated functions for the gain matrices of the parameter-dependent controller

parameter-dependent controller		for $a_y \leq 4.905\ m/s^2$	for $a_y > 4.905\ m/s^2$ $k = p(a_y) = c_1 a_y^4 + c_2 a_y^3 + c_3 a_y^2 + c_4 a_y + c_5$				
		x 10^4	c_1	c_2 x 10^3	c_3 x 10^4	c_4 x 10^5	c_5 x 10^5
$\boldsymbol{K_x}$	$k_{x\,11}$	3.8822	-73.1516	2.1885	-2.3554	1.0330	-1.1707
	$k_{x\,12}$	1.6516	-18.8070	0.5730	-0.6248	0.2687	-0.2169
	$k_{x\,13}$	-4.0111	83.4862	-2.4868	2.6637	-1.1650	1.3555
	$k_{x\,14}$	0.4888	-7.6396	0.2303	-0.2514	0.1136	-0.1310
	$k_{x\,15}$	0.1946	-3.8878	0.1171	-0.1275	0.0573	-0.0703
	$k_{x\,16}$	-0.4663	7.3274	-0.2210	0.2416	-0.1094	0.1269
	$k_{x\,21}$	3.8904	-73.9046	2.2100	-2.3772	1.0421	-1.1831
	$k_{x\,22}$	-1.7716	31.0983	-0.9276	0.9914	-0.4261	0.4421
	$k_{x\,23}$	-3.8162	64.7866	-1.9508	2.1147	-0.9328	1.0329
	$k_{x\,24}$	0.4883	-7.5834	0.2286	-0.2497	0.1129	-0.1299
	$k_{x\,25}$	-0.1937	3.7921	-0.1144	0.1247	-0.0560	0.0685
	$k_{x\,26}$	-0.4672	7.2557	-0.2187	0.2388	-0.1078	0.1237
	$k_{x\,31}$	4.0433	-75.2145	2.2511	-2.4246	1.0651	-1.2076
	$k_{x\,32}$	1.7667	-31.8039	0.9477	-1.0120	0.4349	-0.4562
	$k_{x\,33}$	7.2016	-127.0284	3.8142	-4.1235	1.8172	-2.0378
	$k_{x\,34}$	0.3252	-5.2403	0.1579	-0.1722	0.0776	-0.0898
	$k_{x\,35}$	0.1922	-3.7701	0.1137	-0.1239	0.0557	-0.0682
	$k_{x\,36}$	0.5601	-8.9698	0.2703	-0.2950	0.1329	-0.1533

	$k_{x\,41}$	4.0514	-75.9615	2.2724	-2.4463	1.0742	-1.2199
	$k_{x\,42}$	-1.6295	17.7079	-0.5411	0.5915	-0.2544	0.1977
	$k_{x\,43}$	7.3950	-145.5807	4.3460	-4.6682	2.0475	-2.3579
	$k_{x\,44}$	0.3247	-5.1845	0.1563	-0.1705	0.0769	-0.0887
	$k_{x\,45}$	-0.1930	3.8492	-0.1160	0.1263	-0.0567	0.0695
	$k_{x\,46}$	0.5592	-9.0409	0.2727	-0.2978	0.1344	-0.1564
K_w	$k_{w\,1}$	0.0087	0.0987	-0.0031	0.0037	-0.0021	0.0055
	$k_{w\,2}$	-0.0100	0.0704	-0.0019	0.0016	-0.0003	-0.0017
	$k_{w\,3}$	0.0099	-0.0573	0.0015	-0.0012	0.0001	0.0021
	$k_{w\,4}$	-0.0087	-0.0854	0.0027	-0.0033	0.0019	-0.0051

6.3 Vehicle Handling Test Simulation

The control above has been designed for constant yaw rates and lateral accelerations. In order to see the effectiveness of the designed gain-scheduling control on ride safety and ride comfort, a *double-lane-change* maneuver as one of the standard handling tests for vehicles will be simulated. The path for the maneuver will be generated as optimal path minimizing the lateral acceleration.

6.3.1 Double-lane-change maneuvers

There are two types of double-lane-change maneuvers: one developed by the International Organization for Standardization (ISO) and one introduced by the Consumers Union (CU).

- **ISO 3888-2 double-lane-change maneuver**

The ISO 3888-2 double-lane-change maneuver, which is also known as the “Elk” or “Moose” avoidance test, represents a changing vehicle path based on pre-determined cone placement on the road as illustrated in Figure 6.7.

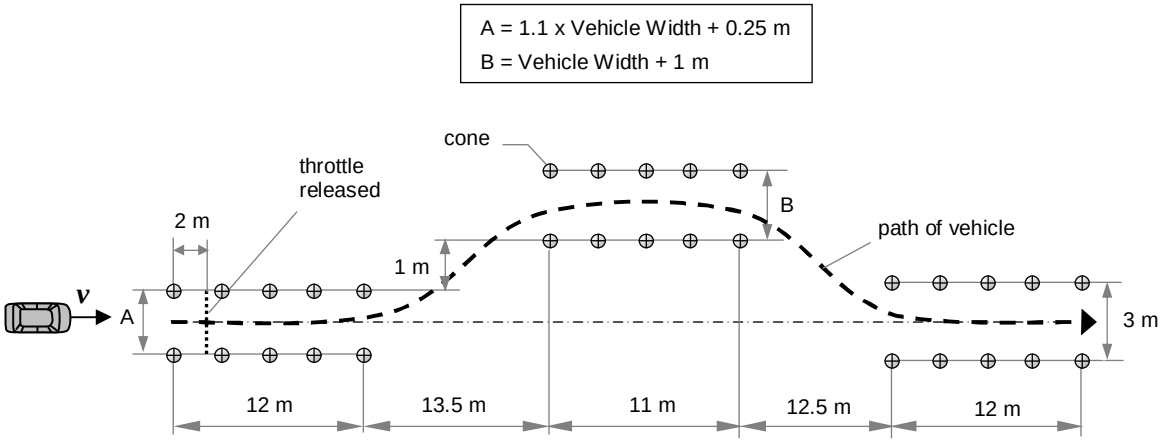

Figure 6.7: ISO 3888-2 double-lane-change maneuver

The widths of the run-up section and swerve section are calculated as functions of the vehicle width. The cone sections must be handled under overrun conditions, i.e. the throttle must be released on entering the section and the car must be in its top gear (or position "D" for vehicles with automatic transmission). The entrance speed is at least $v = 60$ *km/h*. These conditions significantly reduce the possibility of driver influence. The test is considered to be passed if no cones are knocked over, and the maneuver is carried out on a dry road.

- **Consumers Union Short Course double-lane-change maneuver**

The Consumers Union Short Course double-lane-change maneuver is designed to test object avoidance. The schematic of this maneuver is sketched in Figure 6.8. In testing the vehicle is required to exit the original lane to avoid a road obstruction and immediately returns to the original lane without knocking out the cones. Like for the ISO 3888-2 maneuver, the Consumers Union Short Course double-lane-change maneuver is performed on dry road with a minimal entrance speed of $v = 60$ *km/h*.

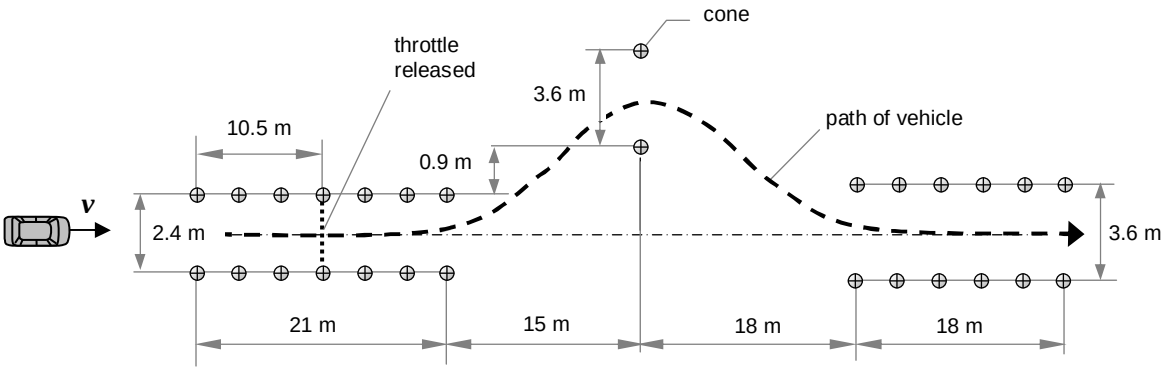

Figure 6.8: Consumers Union Short Course double-lane-change maneuver

6.3.2 Path generation problem for double-lane-change maneuvers

To define the path generation problem for the double-lane-change maneuvers, the path must be modelled first. There are an infinite number of path models that can satisfy the requirements of the ISO 3882-2 and Consumers Union (CU) Short Course double-lane-change maneuvers at a given entrance speed. Aiming to develop a *path model* that is adaptable to a variety of test vehicles and able to produce a repeatable and effectively optimal vehicle path, the paths for the ISO 3882-2 and CU Short Course double-lane-change maneuver are assumed to be composed of a series of straight-line and circular-arc segments with spiral segments between them as illustrated in Figures 6.9 and 6.10, respectively.

The straight-line segments (S) are described by zero curvature, $\kappa = 0$, while constant curvatures $\kappa_j = const.$, $j = 1(1)\ m$, characterize circular-arc segments (C). For the spiral segments, constant rates of curvature changes, $r_k = const.$, $k = 1(1)\ n$, are used. To reduce the complication of the maneuver for test drivers, identical rates of change $| r_k | = r$ can be used. In order to complete the vehicle path, in addition to the curvatures κ_j and the rate of curvatures changes r_k, the time points of changing curvature t_k must be defined. All these parameters are treated as the *design variables* for the path generation problem. The number of design variables is determined by the number m of segments with constant curvatures and the number n of points with changing curvature. For the ISO 3882-2 double-lane-change path we have $m = 4$ and $n = 6$, while we get $m = 3$ and $n = 4$ for the CU Short Course double-lane-change path.

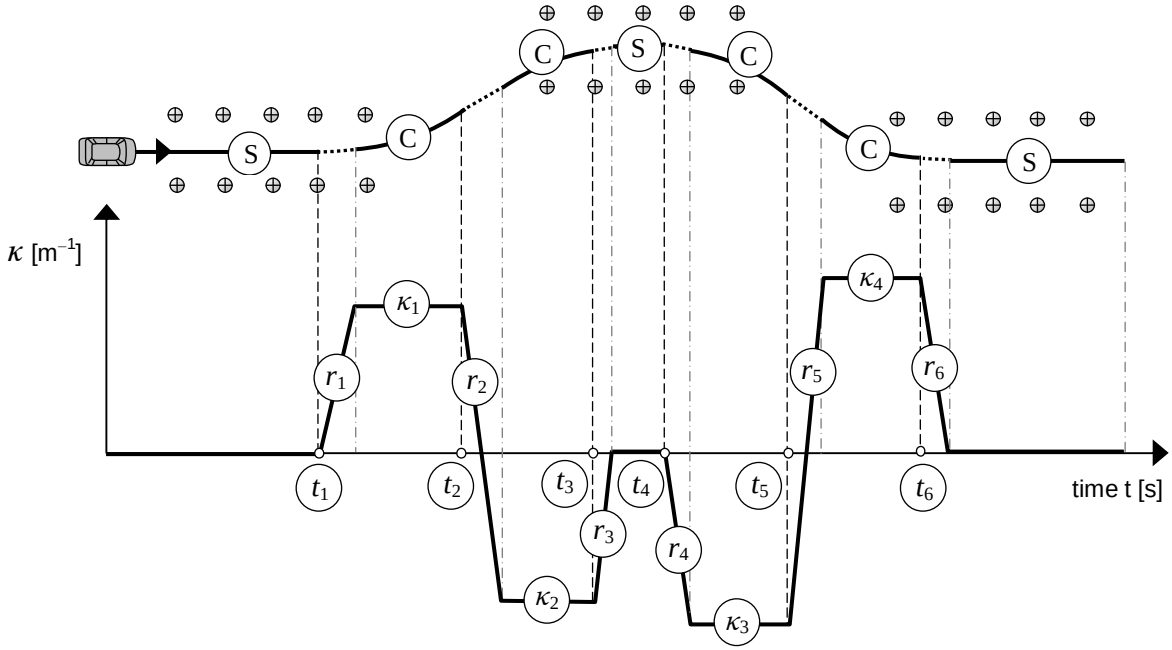

Figure 6.9: Path model for ISO 3882-2 double-lane-change maneuver

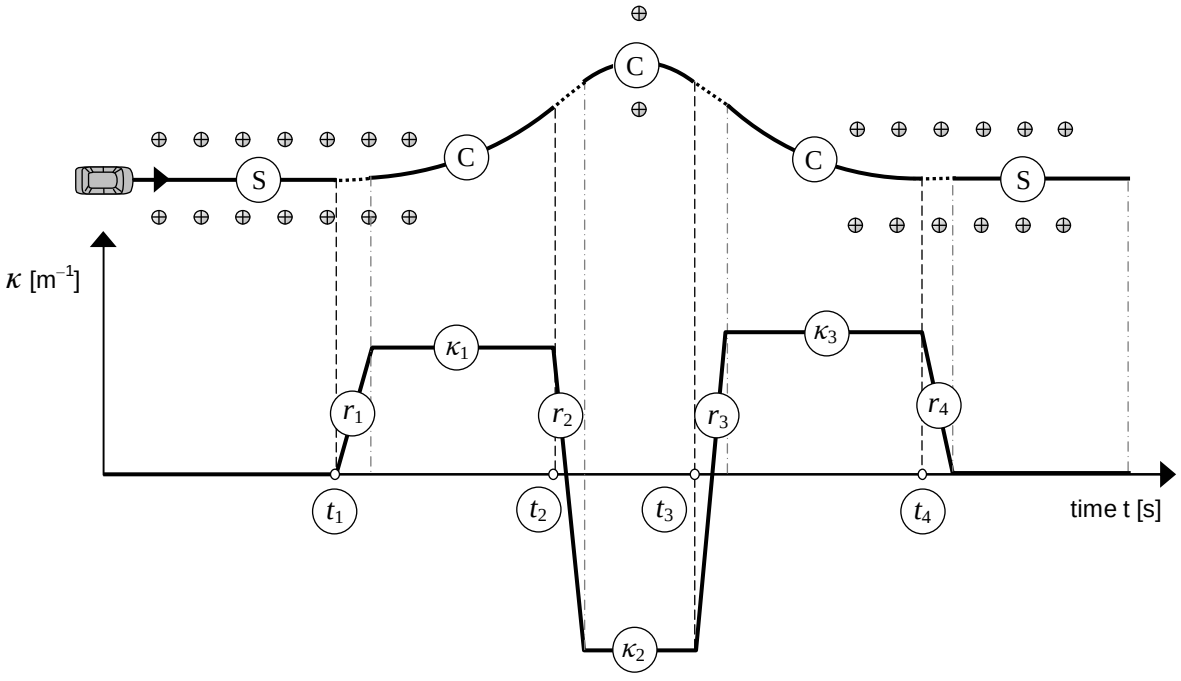

Figure 6.10: Path model for CU Short Course double-lane-change maneuver

The *path constraints* for the ISO 3882-2 and Consumers Union Short Course double-lane-change maneuver are given by the cone positions in Figures 6.7 and 6.8, respectively. To ensure that no cone is knocked out, the tracks of the wheels have to stay within the boundaries generated by the upper and lower cone rows during the cone sections as illustrated in Figure 6.11. Taking into account finite dimensions of tires and cones, the minimal distance between the wheel tracks and the cone center lines can be chosen e.g. as $\delta = 0.30\ m$.

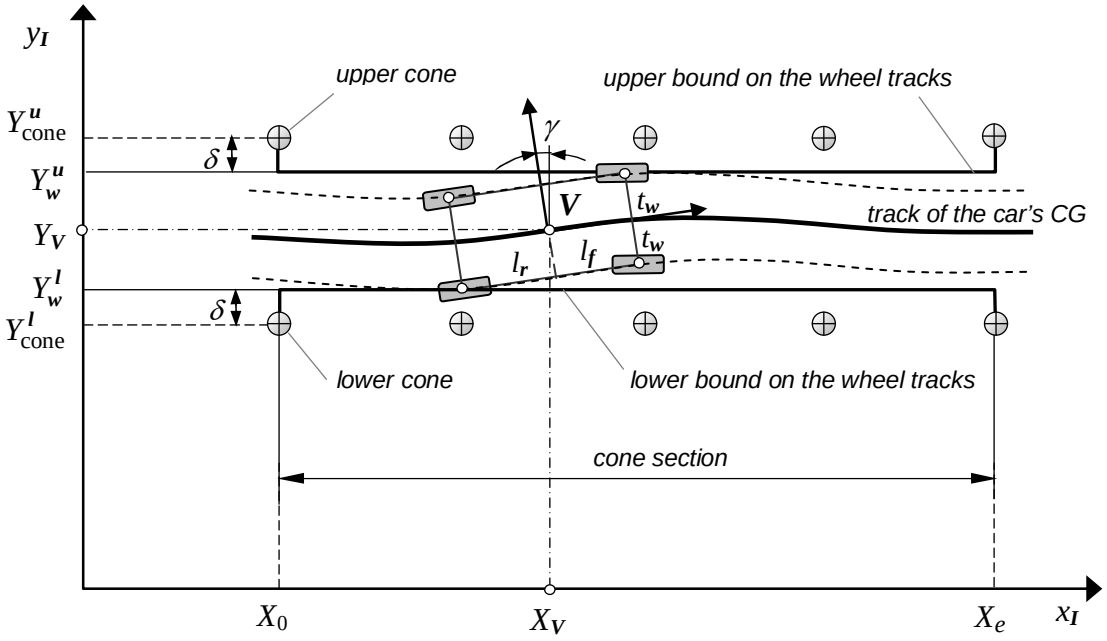

Figure 6.11: Description of the wheel track boundaries in a cone section

The wheel track is a series of the wheel locations (X_{wi}, Y_{wi}) created during simulation, which can be computed from the car's CG location (X_V, Y_V) and rotation matrix defined by yaw angle γ in the absolute inertial reference frame $\boldsymbol{I}$ as

$$\begin{bmatrix} X_{wi} \\ Y_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix} \begin{bmatrix} x_{wi} \\ y_{wi} \end{bmatrix} + \begin{bmatrix} X_V \\ Y_V \end{bmatrix}, \qquad i = 1(1)4 \tag{6.8}$$

where x_{wi} and y_{wi} are the coordinates of the i^{th} wheel in the track coordinate system $\boldsymbol{V}$.

Tables 6.3 and 6.4 show the section dimensions of the ISO 3882-2 and Consumers Union (CU) Short Course double-lane-change (DLC) maneuver and the wheel track boundaries for the studied car where the vehicle width is assumed to be 2.05 *m*. It should be noted that for both types of double-lane-change maneuvers the test requires that the car maintains a straight trajectory only within the first cone section of the course. The car's heading orientation is unrestricted at the end of the test. For none-cone sections, no bounds on the wheel tracks are required. Based on the parameters given in the tables, the constraints on the wheel tracks for the cone sections 1, 3 and 5 can be formulated as shown in Table 6.5.

Table 6.3: Section dimensions and wheel track boundaries for ISO 3882-2 DLC maneuver

position	section 1	section 2	section 3	section 4	section 5
X_0 $[m]$	10.0	22.0	35.5	46.5	59.0
X_e $[m]$	22.0	35.5	46.5	59.0	71.0
Y_{cone}^{u} $[m]$	1.25		5.30		1.50
Y_{cone}^{l} $[m]$	-1.25		2.25		-1.50
Y_w^u $[m]$	0.95		5.00		1.20
Y_w^l $[m]$	-0.95		2.55		-1.20

Table 6.4: Section dimensions and wheel-track boundaries for CU Short Course DLC maneuver

position	section 1	section 2	section 3	section 4	section 5
X_0 $[m]$	10.0	31.0	46.0	46.0	64.0
X_e $[m]$	31.0	46.0	46.0	64.0	82.0
Y_{cone}^{u} $[m]$	1.20		5.70		1.80
Y_{cone}^{l} $[m]$	-1.20		2.10		-1.80
Y_w^u $[m]$	0.90		5.40		1.50
Y_w^l $[m]$	-0.90		2.40		-1.50

Table 6.5: Constraints for the path generation problem

	constraints on the wheel tracks	
	ISO 3882-2 DLC maneuver	CU Short Course DLC maneuver
section 1	$\boldsymbol{h}_1 := \begin{bmatrix} \max\limits_{X_{\boldsymbol{w}1,3} \le 22} \left(Y_{\boldsymbol{w}1,3}\right) - 0.95 \\ -\min\limits_{X_{\boldsymbol{w}2,4} \le 22} \left(Y_{\boldsymbol{w}2,4}\right) - 0.95 \end{bmatrix} \le 0$	$\boldsymbol{h}_1 := \begin{bmatrix} \max\limits_{X_{\boldsymbol{w}1,3} \le 31} \left(Y_{\boldsymbol{w}1,3}\right) - 0.90 \\ \max\limits_{X_{\boldsymbol{w}2,4} \le 31} \left(Y_{\boldsymbol{w}2,4}\right) - 0.90 \end{bmatrix} \le 0$
section 3	$\boldsymbol{h}_2 := \begin{bmatrix} \max\limits_{35.5 \le X_{\boldsymbol{w}1,3} \le 46.5} \left(Y_{\boldsymbol{w}1,3}\right) - 5.00 \\ -\min\limits_{35.5 \le X_{\boldsymbol{w}2,4} \le 46.5} \left(Y_{\boldsymbol{w}2,4}\right) + 2.55 \end{bmatrix} \le 0$	$\boldsymbol{h}_2 := \begin{bmatrix} \max\limits_{X_{\boldsymbol{w}1,3} = 46} \left(Y_{\boldsymbol{w}1,3}\right) - 5.40 \\ -\min\limits_{X_{\boldsymbol{w}2,4} = 46} \left(Y_{\boldsymbol{w}2,4}\right) + 2.40 \end{bmatrix} \le 0$
section 5	$\boldsymbol{h}_3 := \begin{bmatrix} \max\limits_{59 \le X_{\boldsymbol{w}1,3}} \left(Y_{\boldsymbol{w}1,3}\right) - 1.20 \\ -\min\limits_{59 \le X_{\boldsymbol{w}2,4}} \left(Y_{\boldsymbol{w}2,4}\right) - 1.20 \end{bmatrix} \le 0$	$\boldsymbol{h}_3 := \begin{bmatrix} \max\limits_{64 \le X_{\boldsymbol{w}1,3}} \left(Y_{\boldsymbol{w}1,3}\right) - 1.50 \\ -\min\limits_{64 \le X_{\boldsymbol{w}2,4}} \left(Y_{\boldsymbol{w}2,4}\right) - 1.50 \end{bmatrix} \le 0$

In order to avoid vehicle sideway-slipping, the *criterion* for the path generation problem can be defined as the vehicle lateral acceleration. Since for constant speed lateral acceleration is proportional to curvature, see (3.51), minimizing the maximum magnitude of path curvature reduces lateral acceleration arising during the test.

If we restrict ourselves to the case where changes of curvature are the same, $| r_{\boldsymbol{k}} | = r$, the path generation problem can be written as follows:

$$\min_{\boldsymbol{p} \in \mathcal{P}} \max_{j=1(1)\boldsymbol{m}} \left| \kappa_{\boldsymbol{j}}(\boldsymbol{p}) \right|,$$

$$\mathcal{P} := \left\{ \boldsymbol{p} = \left[t_1, \ldots, t_{\boldsymbol{n}}, r, \kappa_1, \ldots, \kappa_{\boldsymbol{m}} \right]^{\boldsymbol{T}} \in \mathbb{R}^{n+m+1} \;\middle|\; \boldsymbol{p}^{\boldsymbol{l}} \le \boldsymbol{p} \le \boldsymbol{p}^{\boldsymbol{u}},\ \boldsymbol{h}_1 \le \boldsymbol{0},\ \boldsymbol{h}_2 \le \boldsymbol{0},\ \boldsymbol{h}_3 \le \boldsymbol{0} \right\}$$

where

- *for the* ISO 3882-2 *double-lane-change path model* in Figure 6.9:

  $m = 4$, $n = 6$,

  $\boldsymbol{p}^{\boldsymbol{l}} = [\ 0.5, 1.0, 1.5, 2.0, 3.0, 3.5, \quad 0.1000, \quad 0.0100, -0.0353, -0.0353, 0.0100]^{\boldsymbol{T}}$,

  $\boldsymbol{p}^{\boldsymbol{u}} = [1.5, 2.0, 2.5, 3.0, 4.0, 4.5, \quad 0.3145, \quad 0.0353, -0.0100, -0.0100, 0.0353]^{\boldsymbol{T}}$,

- *for the* CU Short Course *double-lane-change path model* in Figure 6.10:

  $m = 3$, $n = 4$,

  $\boldsymbol{p}^{\boldsymbol{l}} = [1.0, 2.0, 3.0, 4.0, \quad 0.1000, \quad 0.0100, -0.0353, 0.0100]^{\boldsymbol{T}}$,

  $\boldsymbol{p}^{\boldsymbol{u}} = [3.0, 4.0, 5.0, 6.0, \quad 0.3145, \quad 0.0353, -0.0100, 0.0353]^{\boldsymbol{T}}$. (6.9)

In the above formulation, the upper bound on the rate of curvature change r is defined according to the maximum rate of change for human hand-wheel steering $\delta_{\boldsymbol{st}\,\max} = 720$ *deg/s*, Forkenbrock, et al. [26] and Hac [34], while the upper bound on curvature κ_j is computed from (3.51) with the given car speed $v = 60$ *km/h* and lateral acceleration at the side-slip threshold $a_y^{\boldsymbol{u}} = 9.81\ m/s^2$. The MATLAB.m files used to solve the path generation problem for the ISO 3882-2 double-lane-change maneuver are presented in Appendix D.2 as an example.

During simulation, the entrance speed $v = 60$ *km*/*h* reduces steadily with a constant rate of $dv/dt = -0.5\ m/s^2$. The resulting 11 parameters defining the optimal vehicle path for the ISO 3882-2 double-lane-change maneuver are

$\boldsymbol{t}^* = [\ t_1^*, \ldots, t_6^*]^{\boldsymbol{T}} = [1.0562, \quad 1.6852, \quad 2.4609, \quad 2.5895, \quad 3.2126, \quad 3.9944]^{\boldsymbol{T}}$,

$r = 0.3038$,

$\boldsymbol{\kappa}^* = [\kappa_1^*, \ldots, \kappa_4^*]^{\boldsymbol{T}} = [0.0295, \quad -0.0295, \quad -0.0295, \quad 0.0295]^{\boldsymbol{T}}$.

The path and related information on this maneuver are shown in Figure 6.12.

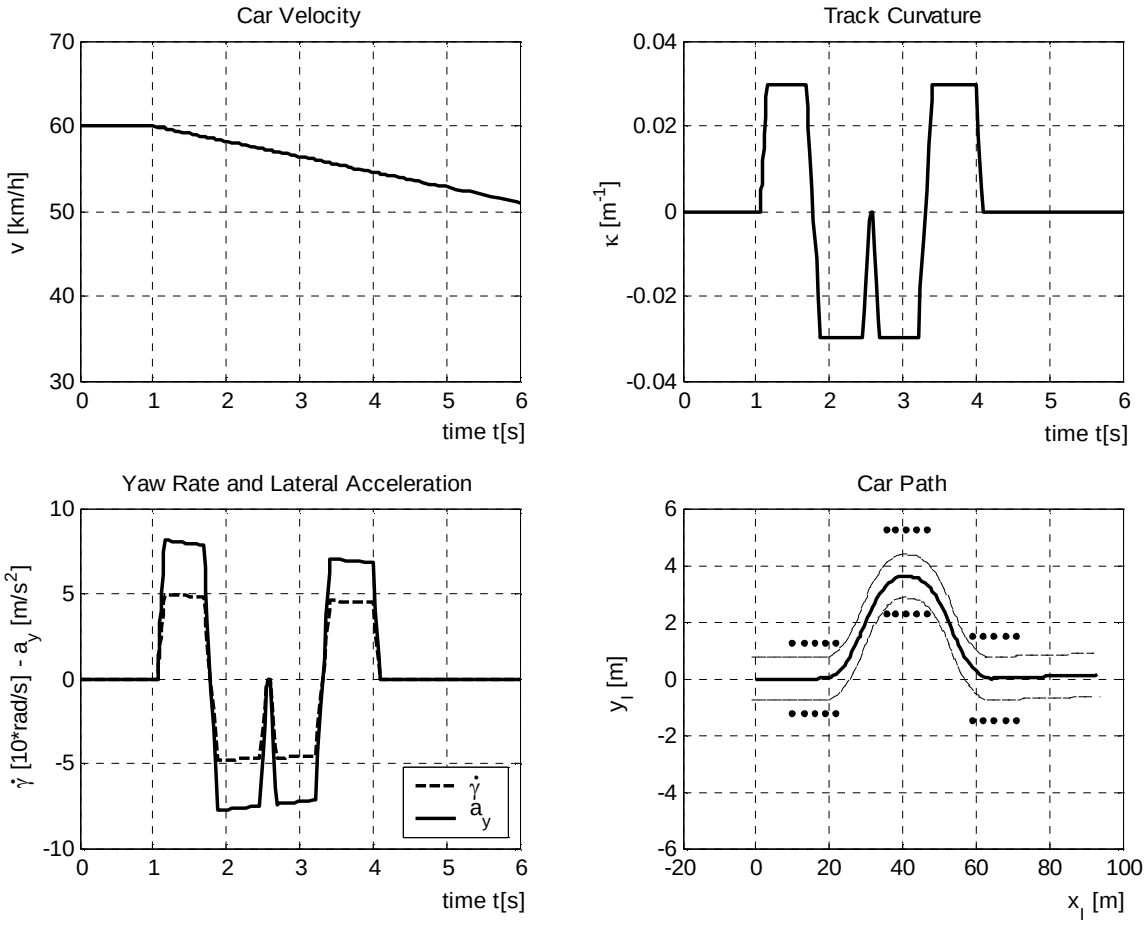

Figure 6.12: Path generation results for ISO 3888-2 DLC maneuver

For the CU Short Course double-lane-change maneuver, the optimal vehicle path is defined by

$$\boldsymbol{t}^* = [\, t_1^*, \ldots, t_4^* \,]^{\boldsymbol{T}} = [1.5327, \quad 2.3258, \quad 3.6428, \quad 4.4464]^{\boldsymbol{T}},$$

$$r = 0.3120,$$

$$\boldsymbol{\kappa}^* = [\, \kappa_1^*, \ldots, \kappa_3^* \,]^{\boldsymbol{T}} = [0.0210, \quad -0.0300, \quad 0.0300]^{\boldsymbol{T}}.$$

The graphical representation of the path is shown in Figure 6.13.

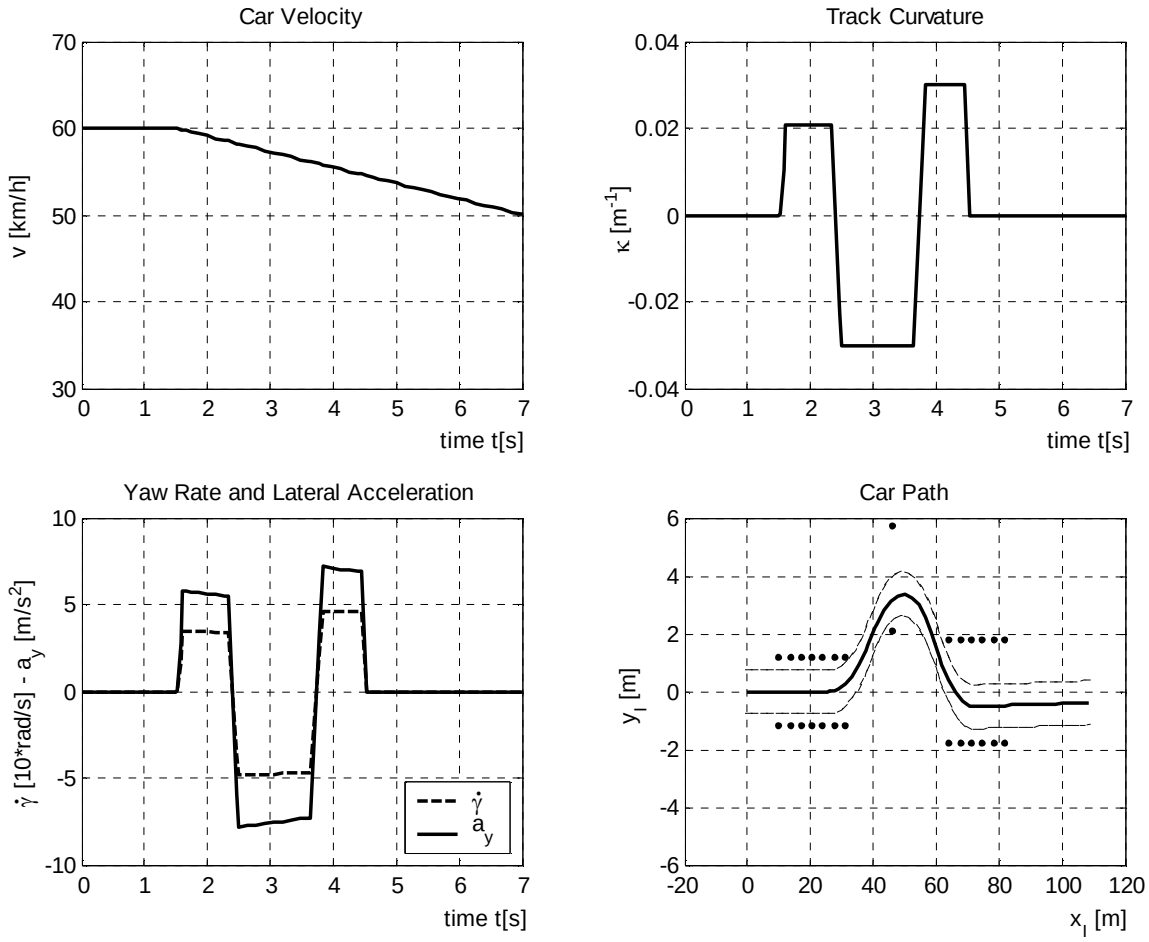

Figure 6.13: Path generation results for CU Short Course DLC maneuver

6.3.3 Simulation results with gain-scheduling control

The optimal path parameters obtained from the path generation problem are used for simulating the studied car in the double-lane-change (DLC) maneuvers with the designed gain-scheduling control. It should be noted that the spatial car model in this case takes on the form of a linear-parameter varying system expressed by (3.76) due to the variation of the car speed and yaw rate during the test.

The simulation results for the ISO 3888-2 and the CU Short Course DLC maneuvers are shown in Figures 6.14 and 6.15, respectively. The system dynamic responses for the optimal passive suspension case are represented by the thin lines, while those for the case of active suspension with gain-scheduling control presented in Table 6.2 are denoted by the thick lines.

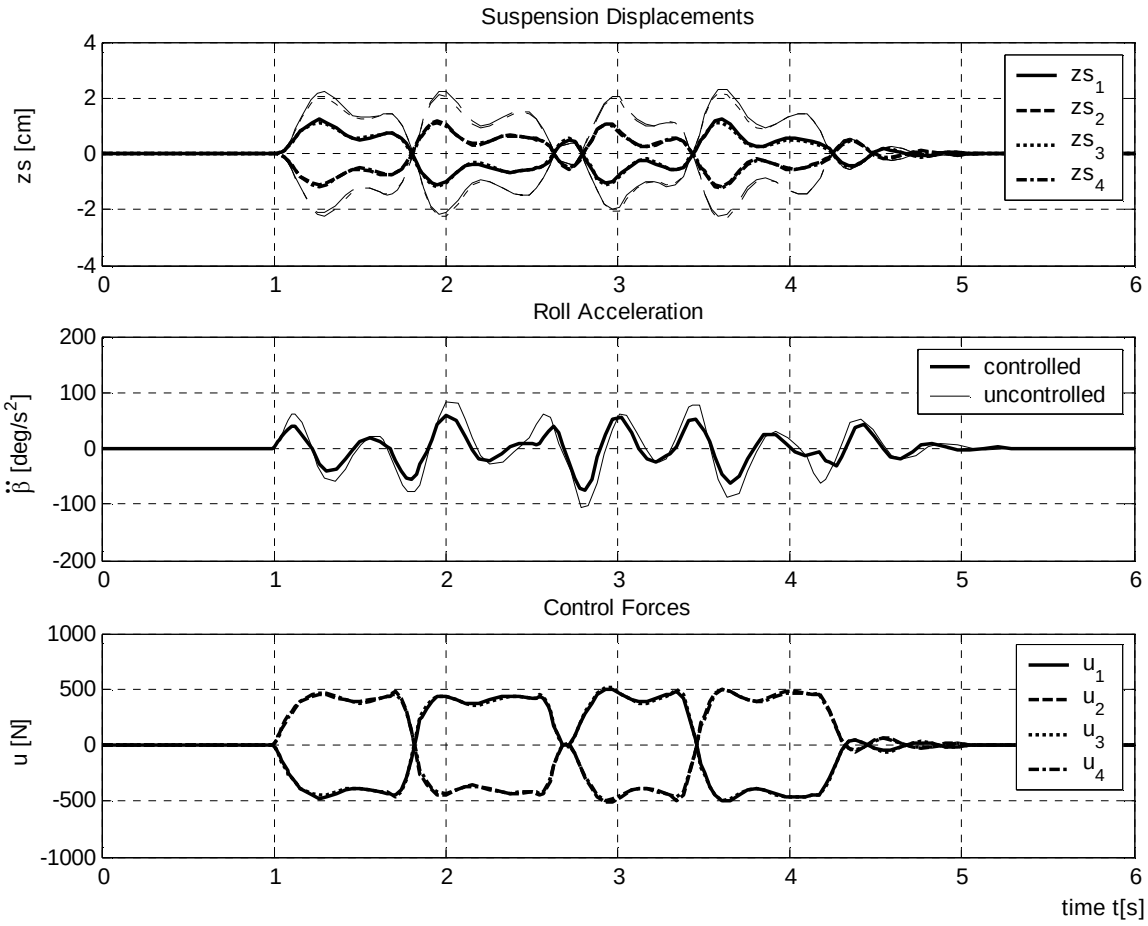

Figure 6.14: Simulation results for ISO 3888-2 DLC maneuver

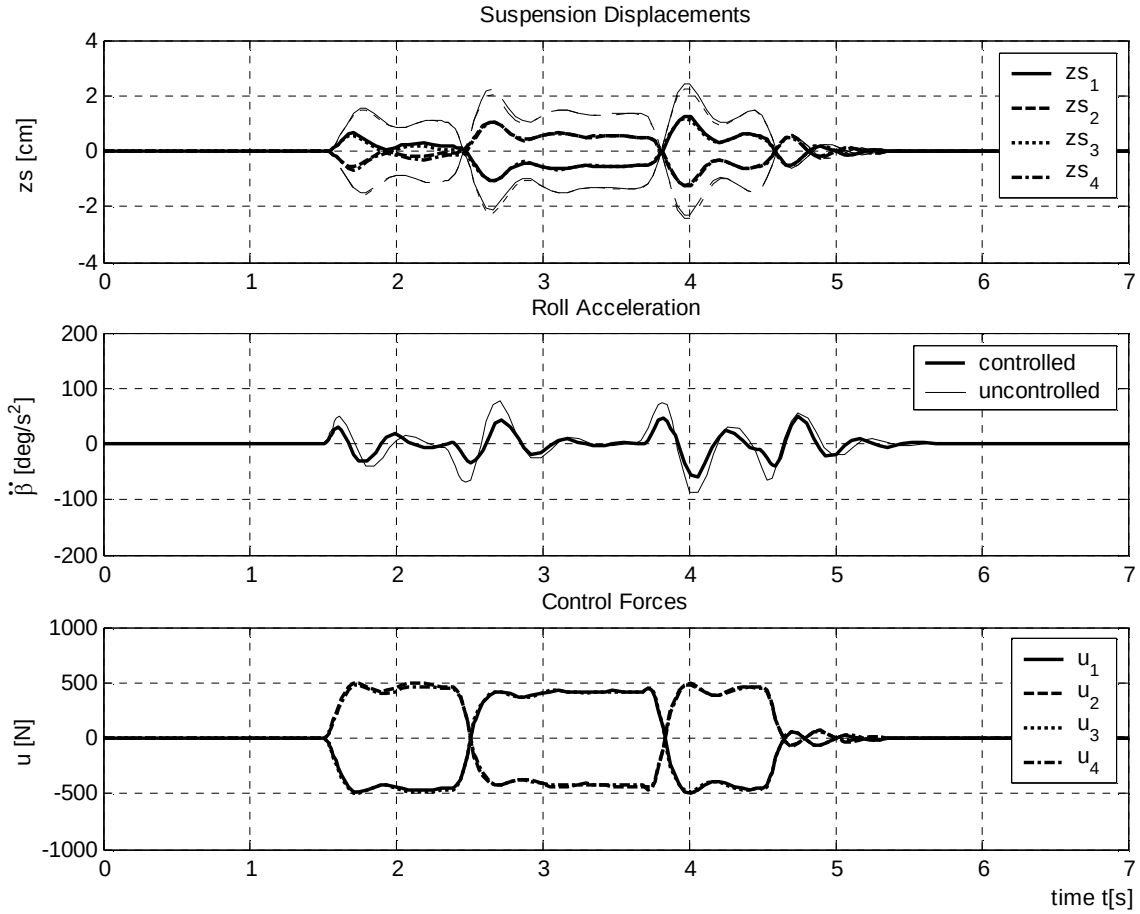

Figure 6.15: Simulation results for CU Short Course DLC maneuver

As can be seen clearly from the figures, the suspension displacements z_{Si} and car body roll acceleration $\ddot{\beta}$ for both double-lane-change maneuvers are significantly reduced by the action of the controlled forces based on the designed parameter-dependent controller. Compared to the case of passive suspension, the active suspension improves the car ride safety by about 47% for the ISO 3888-2 and 52% for the CU Short Course DLC maneuver. At the same time the car ride comfort criterion value reduces by 31% and 38%, respectively. The improvement in the ride comfort criterion of the car can be explained by the optimization results shown in Figure 5.9 where both ride safety and ride comfort criteria tend to lower left corner even when optimizing ride safety only. The obtained results above demonstrate obviously the effectiveness of active suspensions with gain-scheduling control on both ride safety and ride comfort criteria for passenger cars in yaw motion.

Chapter 7

Summary

Aiming to design optimal controllers for passenger cars with active suspensions, optimization methods are applied to various investigations in this dissertation. The obtained results show that both ride safety and ride comfort of the car in yaw motion are improved significantly with the designed controllers. The effectiveness of active suspensions on the vehicle lateral dynamics is demonstrated based on a simplified and linearized three-degree-of-freedom spatial car model featuring handling performance, where the effects of suspension geometry are taken into account.

Since all states and disturbances of the spatial car model are assumed to be available from online measurements, the linear quadratic regulator (LQR) algorithm is applied to define the optimal control law. The LQR control theory is introduced for linear systems without disturbances, and then extended to linear systems with measurable disturbances resulting in an optimal control law with state-feedback and disturbance-feed forward controller. The application of the LQR control on the spatial car model shows the usefulness of this approach since the optimal controllers with a large number of unknowns can be easily derived by choosing appropriately a few weighting factors only.

The obtained results also show that optimal solutions might not be found by choosing the weighting factors by hand if the control problem is subject to constraints on the states, measured outputs or control inputs. This drawback of the conventional method of designing the LQR control is overcome by applying a multi-criterion optimization approach, where the weighting factors become design variables of the associated MCO problem. With the combination of the LQR algorithm the number of design variables of the MCO problem defining the optimal controllers can be reduced significantly. The MCO problem for the spatial car model with active suspension is solved based on the

compromise method and recursive knee approach. Optimization results demonstrate the effectiveness of the proposed optimization method.

The influences of suspension and tire compliance on vehicle rollover stability are considered in order to determine the operating region of the studied spatial car model. Aiming to maintain the desired performances of the system which can be changed due to the variation in the system parameters, gain-scheduling control design is investigated. The global parameter-dependent controller results from the two following design steps: first the local controllers at specified operation points are computed, then the components of the global controller is interpolated as polynomial functions of the system varying parameters. The effectiveness of the designed gain-scheduling control is demonstrated by the simulation of ISO 3882-2 and Consumers Union Short Course double-lane-change maneuvers, which are designated as the standard handling tests for passenger cars. The simulation parameters result from a path which minimizes the vehicle lateral dynamics and is found itself by an optimization problem.

Due to limitation in time, this dissertation cannot focus on all aspects of each spreading problem, and some future research is desirable. To describe sufficiently the vehicle lateral dynamics, the effects of tire dynamics on the vehicle vibration and handling must be included, which were omitted in this research. In addition, although all states and disturbances of the studied spatial car model are measurable, received signals in practice are often corrupted by measurement noise, resulting in discontinuous control. To apply effectively the LQR control, optimal state-feedback and disturbance-feed forward controller has to be combined with a Kalman optimal estimator. Furthermore, the achievements of this research should be verified by practical experiments.

Bibliography

[1] Ackermann, J., *Robuste Regelung - Analyse und Entwurf von linearen Regelungssystemen mit unsicheren physikalischen Parametern*. Berlin: Springer, 1993.

[2] Ammon, D., *Modellbildung und Systementwicklung in der Fahrzeugdynamik*. Stuttgart: Teubner, 1997.

[3] Andersson, J., *Multi-Objective Optimization in Engineering Design*. Ph.D. Thesis. Linköping University, 2001.

[4] Bail, T. R., *A Disturbance-Rejection Problem for a 2-D Airfoil*. M.Sc. Thesis. Virginia University, 1997.

[5] Balas, G., Chiang, R., Packard, A. and Safonov, M., *Robust Control Toolbox User's Guide*, Version 3. The Math Works, Inc., 2004.

[6] Balas, G. J., *Linear Parameter-Varying Control and Its Application to a Turbofan Engine*. In: International Journal of Robust and Nonlinear Control **12** (2002) 763-796.

[7] Bestle, D., *Analyse und Optimierung von Mehrkörpersystemen*. Berlin: Springer, 1994.

[8] Bodie, M. O. and Hac, A., *Closed Loop Yaw Control of Vehicles Using Magneto-Rheological Dampers*. SAE Transactions **109** (2000) 132-139.

[9] Bruzelius, F., *LPV-Based Gain Scheduling*. Ph.D. Thesis. Göteborg: Chalmers University of Technology, 2002.

[10] Chantranuwathana, S. and Peng, H., *Adaptive Robust Force Control for Vehicle Active Suspensions*. International Journal of Adaptive Control and Signal Processing **18** (2004) 83-102.

[11] Chen, P. C. and Huang, A. C., *Adaptive Multiple-Surface Sliding Control of Hydraulic Active Suspension Systems Based on the Function Approximation Technique*. Journal of Vibration and Control **11** (2005) 685-706.

[12] Chen, W., Sahai, A., Messac, A. and Sundararaj, G. J., *Exploration of the Effectiveness of Physical Programming in Robust Design*. ASME Journal of Mechanical Design **122** (2000) 17-26.

[13] Choi, S. B., Lee, H. S. and Park, Y. P., *H_∞ Control Performance of a Full-Vehicle Suspension Featuring Magnetorheological Dampers*. Vehicle System Dynamics **38** (2002) 341-360.

[14] Colaneri, P., Geromel, J. C. and Locatelli, A., *Control Theory and Design*. London: Academic Press Limited, 2001.

[15] Coleman, T., Branch, M. A. and Grace, A., *Optimization Toolbox User's Guide*, Version 3. The MathWorks, Inc., 2004.

[16] Collette, Y. and Siarry, P., *Multi-Objective Optimization*: *Principles and Case Studies*. Berlin: Springer, 2003.

[17] Das, I., *An Improved Technique for Choosing Parameters for Pareto Surface Generation Using Normal-Boundary Intersection*. In: WCSMO-3 Proceedings, Buffalo, (1999) 411-413.

[18] Das, I., *On Characterizing the 'Knee' of the Pareto Curve based on Normal-Boundary Intersection*. Structural and Multidisciplinary Optimization **18** (1999) 107-115.

[19] Das, I. and Dennis, J. E., *Normal-Boundary Intersection*: *A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems*. Society for Industrial and Applied Mathematics **8** (1998) 631-657.

[20] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*. Weinheim: John Wiley & Sons Inc., 2001.

[21] Donahue, M. D., *Implementation of an Active Suspension, Preview Controller for Improved Ride Comfort*. M. Sc. Thesis. University of California at Berkeley, 2001.

[22] Eberhard, P., Dignath, F. and Kübler, L., *Parallel Evolutionary Optimization of Multibody Systems with Application to Railway Dynamics*. Multibody System Dynamics **9** (2003) 143-164.

[23] Eberhard, P., Schiehlen, W. and Bestle, D., *Some Advantages of Stochastic Methods in Multicriteria Optimization of Multibody Systems*. Archive of Applied Mechanics **69** (1999) 543-554.

[24] Eberhard, P. and Schiehlen, W., *Hierarchical Modeling in Multibody Dynamics*. Archive of Applied Mechanics **68** (1998) 237-246.

[25] Fitzpatrick, K. L., *Applications of Linear Parameter-Varying Control for Aerospace Systems*. M. Sc. Thesis. University of Florida, 2003.

[26] Forkenbrock, G. J., Elsasser, D. and O'Harra, B., *NHTSA's Light Vehicle Handling and ESC Effectivness Research Program*. URL http://www-nrd.nhtsa.dot.gov/pdf/nrd-01/esv/ esv19/05-0221-O.pdf, 2005.

[27] Fujiwara, Y. and Adachi, S., *Steering Assistance System for Driver Characteristics using Gain Scheduling Control*. URL http://arx.ee.utsunomiya-u.ac.jp/English/papers/abst2003/ECC2003.pdf, 2003.

[28] Gahinet, P., Nemirovski, A., Laub, A. J. and Chilali, M., *LMI Control Toolbox User's Guide*. Version 1. The MathWorks, Inc., 1995.

[29] Gaspar, P., Szaszi, I. and Bokor, J., *Active Suspension Design Using Linear Parameter Varying Control*. International Journal of Vehicle Autonomous Systems **1** (2003) 206-221.

[30] Gärtner, A. and Saeger, M., *Simulationsumgebung zur Untersuchung aktiver Wankstabilisierung in Verbindung mit einer Fahrdynamikregelung*. Essen: Simulation in der Fahrzeugdynamik, 2003.

[31] Genç, S., *Synthesis and Properties of Magnetorheological (MR) fluids*. Ph. D. Thesis. University of Pittsburgh, 2002.

[32] Genc, A. U., *Linear Parameter Varying Modelling and Robust Control of Variable Cam Timing Engines*. Ph. D. Thesis. University of Cambridge, 2002.

[33] Giua, A., Seatzu, C. and Usai, G., *A Mixed Suspension System for a Half-Car Vehicle Model*. Dynamics and Control **10** (2000) 375-397.

[34] Hac, A., *Influence of Active Chassis Systems on Vehicle Propensity to Maneuver-Induced Rollovers*. SAE Paper 2002-01-0967, 2002.

[35] Hac, A., *Rollover Stability Index Including Effects of Suspension Design*. SAE Paper 2002-01-0965, 2002.

[36] Halfmann, C. and Holzmann, H., *Adaptive Modelle für die Kraftfahrzeugdynamik*. Berlin: Springer, 2003.

[37] He, Y., *Design of Rail Vehicles with Passive and Active Suspensions Using Multidisciplinary Optimization, Multibody Dynamics, and Genetic Algorithms*. Ph. D. Thesis. University of Waterloo, 2003.

[38] Hong, K., Sohn, H. C., and Hedrick, J. K., *Modified Skyhook Control of Semi-Active Suspensions*: *A New Model, Gain Scheduling, and Hardware-in-the-Loop Tuning*. ASME Journal of Dynamic Systems, Measurement and Control **124** (2002) 158-167.

[39] Hyvärinen, J. P., *The Improvement of Full Vehicle Semi-Active Suspension through Kinematical Model*. Ph.D. Thesis. University of Oulu, 2004.

[40] Kiencke, U. and Nielsen, L., *Automotive Control Systems*. Berlin: Springer, 2000.

[41] Kim, I. Y. and Weck, O. L. D., *Adaptive Weighted-Sum Method for Bi-Objective Optimization*: *Pareto Front Generation*. Structural and Multidisciplinary Optimization **29** (2004) 149-158.

[42] Kortüm, W. and Lugner, P., *Systemdynamik und Regelung von Fahrzeugen*. Berlin: Springer, 1994.

[43] Krüger, W., *Integrated Design Process for the Development of Semi-Active Landing Gears for Transport Aircraft*. Ph.D. Thesis. University of Stuttgart, 2000.

[44] Kumar, A., *Convex Modeling Techniques for Aircraft Control*. M.Sc. Thesis. Virginia Polytechnic Institute and State University, 2000.

[45] Lazic, N., *Optimal Vehicle Dynamics - Yaw Rate and Side Slip Angle Control Using 4-Wheel Steering*. Ph.D. Thesis. Lund Institute of Technology, 2002.

[46] Lu, G., Huang, J., and Tomizuka, M., *Vehicle Lateral Control Under Fault in Front and/or Rear Sensors*. University of California: PATH Research Report, UCB-ITS-PRR-2003-26, 2003.

[47] Marler, R.T. and Arora, J.S., *Survey of Multi-Objective Optimization Methods for Engineering*. Structural and Multidisciplinary Optimization **26** (2004) 369-395.

[48] Mattson, C. A., Mullur, A. A., and Messac, A., *Obtaining a Minimal Representation of Multi-Objective Design Space*. Engineering Optimization **36** (2004) 721-740.

[49] Messac, A., Yahaya, A. I., and Mattson, C. A., *The Normalized Normal Constraint Method for Generating the Pareto Frontier*. Structural and Multidisciplinary Optimization **25** (2003) 86-98.

[50] Mitschke, M., *Dynamik der Kraftfahrzeuge*, Band B: Schwingungen. Berlin: Springer, 1990.

[51] Mitschke, M., *Dynamik der Kraftfahrzeuge*, Band C: Fahrverhalten. Berlin: Springer, 1990.

[52] Moscinski, J. and Ogonowski, Z., *Advanced Control with Matlab and Simulink*. London: Ellis Horwood Limited, 1995.

[53] Moylan, P. J. and Moore, J. B., *Generalizations of Singular Optimal Control Theory*. London: Pergamon Press, 1971.

[54] O'Hara, S. R., *Vehicle Path Optimization of Emergency Lane Change Maneuvers for Vehicle Simulation.* M. Sc. Thesis. University of Maryland, 2005.

[55] Pang, L., Kamath, G. M., and Wereley, N. M., *Analysis and Testing of a Linear Stroke Magnetorhehological Damper*. AIAA/ASME/AHS Adaptive Structures **4** (1998) 2841-2856.

[56] Paré, C. A., *Experimental Evaluation of Semiactive Magneto-Rheological Suspensions for Passenger Vehicles*. M. Sc. Thesis. Virginia Polytechnic Institute and State University, 1988.

[57] Park, J. H. and Kim, Y. S., *Decentralized Variable Structure Control for Active Suspensions based on a Full-Car Model*. In: Proceedings of the International Conference on Control Applications, Trieste (1998) 383-387.

[58] Petersen, I., Johansen, T. A., Kalkkuhl, J., and Lüdermann, J., *Wheel Slip Control in ABS Brakes Using Gain Scheduled Constrained LQR*. In: Control Systems Technology **11** (2003) 799-811.

[59] Popp, K. and Schiehlen, W., *Fahrzeugdynamik*. Stuttgart: Teubner, 1993.

[60] Rahnejat, H., *Multi Body Dynamics – Vehicles, Machines and Mechanisms*. Warrendale: Society of Automotive Engineers, 1998.

[61] Ramirez, W. F., *Process Control and Identification*. San Diego: Academic Press Inc., 1994.

[62] Rettig, U. and Stryk, O., *Optimal and Robust Damping Control for Semi-Active Vehicle Suspension*. In: Proceedings of the Fifth EUROMECH Nonlinear Dynamics Conference, Eindhoven University of Technology, MS 20-316, 2005.

[63] Rill, G., *Vehicle Modelling for Real Time Applications*. Journal of the Braz. Soc. Mechanical Sciences **19** (1997) 192-206.

[64] Rouieh, S. and Titli, A., *Design of Active and Semi-Active Automotive Suspension Using Fuzzy Logic*. In: Proceedings of the IFAC World Congress, Sydney (1993) 253-257.

[65] Ryu, J., *State and Parameter Estimation for Vehicle Dynamics Control Using GSP*. Ph. D. Thesis. Stanford University, 2004.

[66] Sampson, D. J. M. and Cebon, D., *Active Roll Control of Single Unit Heavy Road Vehicles*. Vehicle System Dynamics **40** (2003) 229-270.

[67] Sampson, D. J. M., *Active Roll Control of Articulated Heavy Vehicles*. Ph.D. Thesis. University of Cambridge, 2000.

[68] Sastry, S. and Bodson, M., *Adaptive Control - Stability, Convergence, and Robustness*. New Jersey: Prentice-Hall, 1989.

[69] Schiehlen, W., *Technische Dynamik*: *Eine Einführung in die analytische Mechanik und ihre technischen Anwendungen*. Stuttgart: Teubner, 1986.

[70] Schiehlen, W., *Multibody System Dynamics*: *Roots and Perspectives*. Multibody System Dynamics **1** (1997) 149-188.

[71] Schwarz, H., *Theoretische und experimentelle Methoden der Regelungstechnik*, Band 12: Optimale Regelung lineare Systeme. Zürich: Bibliographisches Institut AG, 1976.

[72] Shahian, B. and Hassul, M., *Control System Design Using Matlab*. New Jersey: Prentice-Hall International, Inc., 2001.

[73] Shen, X. and Peng, H., *Analysis of Active Suspension Systems with Hydraulic Actuators*. Proceedings of the IAVSD Conference, Atsugi, **2** (2003) 546-556.

[74] Shukla, P. K. and Deb, K., *On Finding Multiple Pareto-Optimal Solutions Using Classical and Evolutionary Generating Methods*. KanGAL Report Number 2005006, Indian Institute of Technology Kanpur, 2005.

[75] Siouris, G. M., *Optimal Control and Estimation Theory*. New York: John Wiley & Sons, Inc., 1996.

[76] Spencer, B. F., Dyke, S. J., Sain, M. K. and Carlson, J. D., *Phenomenological Model of a Magnetorheological Damper*. ASCE Journal of Engineering Mechanics **3** (1996) 201-221.

[77] Steinberg, P., *Fahrzeugtechnik 1&2*. Lecture notes. Brandenburg University of Technology Cottbus, 2005.

[78] Taghirad, H. D. and Esmailzadeh, E., *Automobile Passenger Comfort Assured Through LQG/LQR Active Suspension*. Journal of Vibration and Control **4** (1998) 603-618.

[79] Ungoren, A. Y. and Peng, H., *An Adaptive Lateral Preview Driver Model*. Vehicle System Dynamics **43** (2005) 245 - 259.

[80] Ungoren, A. Y. and Peng, H., *Evaluation of Vehicle Dynamic Control for Rollover Prevention*. International Journal of Automotive Technology **5** (2004) 115-122.

[81] Ungoren, A. Y., Peng, H. and Milot, D. R., *Rollover Propensity Evaluation of an SUV Equipped with a TRW VSC System*. SAE Paper 2001-01-0128, 2001.

[82] Valasek, M., Novak, M., Sika, Z. and Vaculın, O., *Extended Ground-Hook - New Concept of Semi-Active Control of Truck's Suspension*. Vehicle System Dynamics **27** (1997) 289-303.

[83] Vaughan, J. E., *Active and Semi-Active Control to Counter Vehicle Payload Variation*. M.Sc. Thesis. Georgia Institute of Technology, 2004.

[84] Venhovens, P. J. T. and Nabb, K., *Vehicle Dynamics Estimation Using Kalman Filters*. Vehicle System Dynamics **32** (1999) 171-184.

[85] Viano, D. C. and Parenteau, C., *Case Study of Vehicle Maneuvers Leading to Rollovers*: *Need for a Vehicle Test Simulating Off-Road Excursions, Recovery and Handling*. SAE Paper 2003-01-0169, 2003.

[86] Wachal, A. and Bestle, D., *Use of Air Spring Elements for Vibration Insulation*. In: Proceedings of the Fifth EUROMECH Nonlinear Dynamics Conference, Eindhoven University of Technology, MS 03-295, 2005.

[87] Wallentowitz, H., *Vertikal-/Querdynamik von Kraftfahrzeugen*. Institut für Kraftfahrtwesen Aachen, 2000.

[88] Wang, J. Y. and Tomizuka, M., *Dynamic Analyses and Robust Lateral Control Design for Automated Lane Guidance of Heavy Vehicles.* Asia Journal of Control **2** (2000) 341-369.

[89] Wang, J. Y. and Tomizuka, M., *Gain-Scheduled H-infinity Loop-Shaping Control for Automated Lane Guidance of Tractor-Semitrailer Combination Vehicles.* In: Proceedings of American Control Conference, Chicago (2000) 2033-2037.

[90] Wang, J. Y. and Tomizuka, M., *Robust H_∞ Lateral Control of Heavy-Duty Vehicles in Automated Highway System.* In: Proceedings of the American Control Conference, San Diego, **5** (1999) 3671-3675.

[91] Wang, Y., Hartmann, T. S., Schinkel, M. and Hunt, K. J., *A New Approach to Simultaneous Stabilisation with D Stability and Its Application to Control of Antilock Braking Systems*. In: Proceedings of the International Mechanical Engineering Congress and Exposition, Anaheim (2001) 729-735.

[92] Willumeit, H. P., *Modelle und Modellierungsverfahren in der Fahrzeugdynamik*. Stuttgart: Teubner, 1998.

[93] Wu, F., *Control of Linear Parameter Varying Systems*. Ph.D. Thesis. University of California at Berkeley, 1995.

[94] Yi K. and Song B. S., *Observer Design for Semi-Active Suspension Control*. Vehicle System Dynamics **32** (1999) 129-148.

[95] Yokoyama, M., Hedrick, J. K. and Toyama, S., *A Model Following Sliding Mode Controller for Semi-Active Suspension Systems with MR Dampers.* In: Proceedings of the American Control Conference, Arlington (2001) 2652-2657.

[96] Zhong, Z., *Study on Control Algorithms for Semi-Active Suspensions*. Research Report. Cottbus: Brandenburg University of Technology, 2003.

Appendix A

Parameters of the Spatial Car Model

This thesis focuses on a spatial car model introduced in Chapter 3. For computations and simulations, the parameters in Table A.1 are used throughout this dissertation.

Table A.1: Parameters of the studied passenger car

	parameter	**notation**	**value**	**unit**
1	sprung mass	m_S	1460.0	kg
2	roll moment of inertia of the sprung mass	I_X	460.0	$kg\ m^2$
3	pitch moment of inertia of the sprung mass	I_Y	2460.0	$kg\ m^2$
4	yaw moment of inertia of the sprung mass	I_Z	1900.0	$kg\ m^2$
5	front suspension damping rate	b_f	1290.0	Ns/m
6	rear suspension damping rate	b_r	1620.0	Ns/m
7	front suspension stiffness	k_f	19960.0	N/m
8	rear suspension stiffness	k_r	17500.0	N/m
9	modified front anti-roll bar stiffness	r_f	19200.0	$N/m/rad$
10	modified rear anti-roll bar stiffness	r_r	9600.0	$N/m/rad$

11	vertical front tire stiffness	k_{tf}	175500.0	*N/m*
12	vertical rear tire stiffness	k_{tr}	175500.0	*N/m*
13	distance from roll axis to the car's CG	h_{RC}	0.2	*m*
14	height of roll axis above ground	h_{RV}	0.3	*m*
15	half the distance between the front wheels	t_f	0.761	*m*
16	half the distance between the rear wheels	t_r	0.755	*m*
17	distance from the car's CG to front axle	l_f	1.011	*m*
18	distance from the car's CG to rear axle	l_r	1.803	*m*
19	minimum turning radius	$R_{\min}$	10.0	*m*
20	maximum car speed	$v_{\max}$	56	*m/s*

Appendix B

NEWEUL Output File for the Car Model

NEWEUL is a FORTRAN program for computer-aided generation of symbolic equations of motion of a multi-body system. This program can be applied in various technical fields, such as vehicle dynamics, robotics, mechanism dynamic analysis, and for any topological structure of multi-body systems with holonomic or non-holonomic constraints. The equations of motion of a multi-body system in both nonlinear and linearized forms can be automatically generated by the formalisms described in Chapter 2. The output file of NEWEUL presents not only the results, but also the contents of the input file which has to be provided by the user. In the following, the NEWEUL output file obtained for the three degree-of-freedom spatial car model is shown.

```
C>   RAEUMLICHES FAHRZEUGMODELL
C>
C>   Tuan-Anh. Nguyen       September,2003
C>
C>=======================================
C>
     0     Steuerparameter: 0=holonom, 1=nichtholonom
     0     Komprimierungsart(0-6): 0=vollsymbolisch
     4     Zahl der Koordinatensysteme
     3     Zahl der (Lage-)Freiheitsgrade
     3     Zahl der Hilfsvarablen
     9     Zahl der linearisierbaren Groessen
     0     Zahl der numerischen Groessen
     2     Zahl der Vereinfachungen
     0     Zahl der Substitutionsvariable
    /NEGST=2/
C>        (Darstellung der Newton-Eulerschen Gleichungen
C>          im Inertialsystem)
C>
C>
C>    LAGEVEKTOR
C>    ************
     Y:     ZC
     Y:     BETA
     Y:     ALPHA
C>
     2     Automatische Ableitung (1: 1/2; 2: P/PP; 0: keine)
C>
C>
C>          HILFSVARIABLEN
C>          **************
C>
     HV:    XV(T)
     HV:    YV(T)
     HV:    GAMMA(T)
C>
C>
C>    LINEARISIERUNG
C>    ****************
C>
     1    Vollstaendige Linearisierung
C>
C>
C>          VEREINFACHUNGEN
```

```
C>
      VFV: XVOTOT*cos(GAMMA)+YVOTOT*sin(GAMMA)
      VFE: AX
C>
      VFV:-XVOTOT*sin(GAMMA)+YVOTOT*cos(GAMMA)
      VFE: AY
C>
C>***********************************************
C>***                                         ***
C>***          A u s g a b e b l o c k  1     ***
C>***                                         ***
C>***       A l l g e m e i n e  A n g a b e n ***
C>***                                         ***
C>***********************************************
C>
C>  Lagevektor
      Y(1)=ZC
      Y(2)=BETA
      Y(3)=ALPHA
C>
C>  1. Ableitung des Lagevektors
      Y1(1)=ZCP
      Y1(2)=BETAP
      Y1(3)=ALPHAP
C>
C>  2. Ableitung des Lagevektors
      Y2(1)=ZCPP
      Y2(2)=BETAPP
      Y2(3)=ALPHAPP
C>
C>  Hilfsvariable
C>
      XV=XV(T)
C>
      YV=YV(T)
C>
      GAMMA=GAMMA(T)
C>
C>  Linearisierbare Groessen
C>
      ALPHAPP     BETAPP      ZCPP        ALPHAP
      BETAP       ZCP         ALPHA       BETA
      ZC
C>
C>  Vereinfachungen
C>
      VFV=XVOTOT*COS(GAMMA)+YVOTOT*
          SIN(GAMMA)
      VFE=AX
C>
      VFV=-XVOTOT*SIN(GAMMA)+YVOTOT*
          COS(GAMMA)
      VFE=AY
C>
C>***********************************************
C>***                                         ***
C>***       A u s g a b e b l o c k  1 - E n d e ***
C>***                                         ***
C>***********************************************
C>     KOORDINATENSYSTEME
C>     ************************
C>
C>     Vehicle Fixed Coordinate - Track Motion
C>          ------------------------------------------
      KOSART:     R     - Referenzsystem
      KOSYNA:     YA    - Namen des Koordinatensystems
      KOSYNA:     I     - Namen des Bezugssystems
C>
C>     ****  Rotatorischer Teil  ****
      1     Zahl der Teildrehungen
C>
      3     Art der Drehung (-3_+5)
C>
C>     Drehwinkel
      WINK=     GAMMA
C>
C>     ****  Translatorischer Teil  ****
      1     Zahl der Teilvektoren
C>
      KOSYNA:     I               System fuer Teilvektor
C>
C>     Teilvektor
      R(1)= XV
      R(2)= YV
      R(3)= 0
C>
C>*******************************************
C>
C>     Roll Motion
C>     ---------------------------
      KOSART:     R     - Referenzsystem
      KOSYNA:     RO    - Namen des Koordinatensystems
      KOSYNA:     YA    - Namen des Bezugssystems
C>
C>     ****  Rotatorischer Teil  ****
      2     Zahl der Teildrehungen
C>
      2     Art der Drehung (-3_+5)
C>
C>     Drehwinkel
      WINK =     THETA
C>
      1     Art der Drehung (-3_+5)
C>
C>     Drehwinkel
      WINK =     BETA
C>
C>     ****  Translatorischer Teil  ****
      1     Zahl der Teilvektoren
C>
      KOSYNA:     YA              System fuer Teilvektor
C>
C>     Teilvektor
      R(1)= 0
      R(2)= 0
      R(3)= HRV+ZC
C>
C> *****************************************
```

```
C>
C>     Car Center
C>     ----------
     KOSART:     R     - Referenzsystem
     KOSYNA:     PI    - Namen des Koordinatensystems
     KOSYNA:     RO    - Namen des Bezugssystems
C>
C>     ****  Rotatorischer Teil  ****
     1     Zahl der Teildrehungen
C>
     -2    Art der Drehung (-3_+5)
C>
C>      Drehwinkel
     WINK =      THETA
C>
C>     ****  Translatorischer Teil  ****
     1     Zahl der Teilvektoren
C>
     KOSYNA:     PI                System fuer Teilvektor
C>
C>      Teilvektor
     R(1)= 0
     R(2)= 0
     R(3)= HRC
C>
C> ***********************************************
C>
C>     Car with Pitch Motion
C>     -------------------------------------
     KOSART:     S         - Starrkörper
     KOSYNA:     CAR       - Namen des Koordinatensystems
     KOSYNA:     PI        - Namen des Bezugssystems
C>
C>     ****  Rotatorischer Teil  ****
     1     Zahl der Teildrehungen
C>
     2     Art der Drehung (-3_+5)
C>
C>      Drehwinkel
     WINK =      ALPHA
C>
C>     ****  Translatorischer Teil  ****
     0     Zahl der Teilvektoren
C>
C>*********************************************
C>***                                       ***
C>***            A u s g a b e b l o c k  2  ***
C>***                                       ***
C>***         K o o r d i n a t e n s y s t e m e  ***
C>***                                       ***
C>*********************************************
C>
C>   Drehungsmatrix YA bzgl. Inertialsystem
     DS1(1,1)=COS(GAMMA)
     DS1(1,2)=-SIN(GAMMA)
     DS1(1,3)=0.
     DS1(2,1)=SIN(GAMMA)
     DS1(2,2)=COS(GAMMA)
     DS1(2,3)=0.
     DS1(3,1)=0.
     DS1(3,2)=0.
     DS1(3,3)=1.
     C>   Ortsvektor YA bzgl. Inertialsystem
     DR1(1)=XV
     DR1(2)=YV
     DR1(3)=0.
C>
C>   Drehungsmatrix RO bzgl. YA
     DS2(1,1)=COS(THETA)
     DS2(1,2)=SIN(THETA)*SIN(BETA)
     DS2(1,3)=SIN(THETA)*COS(BETA)
     DS2(2,1)=0.
     DS2(2,2)=COS(BETA)
     DS2(2,3)=-SIN(BETA)
     DS2(3,1)=-SIN(THETA)
     DS2(3,2)=SIN(BETA)*COS(THETA)
     DS2(3,3)=COS(THETA)*COS(BETA)
C>
C>   Ortsvektor RO bzgl. YA
     DR2(1)=0.
     DR2(2)=0.
     DR2(3)=HRV+ZC
C>
C>   Drehungsmatrix PI bzgl. RO
     DS3(1,1)=COS(THETA)
     DS3(1,2)=0.
     DS3(1,3)=-SIN(THETA)
     DS3(2,1)=0.
     DS3(2,2)=1.
     DS3(2,3)=0.
     DS3(3,1)=SIN(THETA)
     DS3(3,2)=0.
     DS3(3,3)=COS(THETA)
C>
C>   Ortsvektor PI bzgl. RO
     DR3(1)=-HRC*SIN(THETA)
     DR3(2)=0.
     DR3(3)=HRC*COS(THETA)
C>
C>   Drehungsmatrix CAR bzgl. PI
     DS4(1,1)=COS(ALPHA)
     DS4(1,2)=0.
     DS4(1,3)=SIN(ALPHA)
     DS4(2,1)=0.
     DS4(2,2)=1.
     DS4(2,3)=0.
     DS4(3,1)=-SIN(ALPHA)
     DS4(3,2)=0.
     DS4(3,3)=COS(ALPHA)
C>
C>   Ortsvektor CAR bzgl. PI
     DR4(1)=0.
     DR4(2)=0.
     DR4(3)=0.
C>
C>*********************************************
C>***                                       ***
C>***        A u s g a b e b l o c k  2 - E n d e     ***
C>***                                       ***
C>*********************************************
```

```
C>***          A u s g a b e b l o c k  5                ***
C>***                                                   ***
C> K i n e m a t i s c h e  G r o e s s e n  Referenzsystem
C>***                                                   ***
C>*************************************************
C>  Kinematische Groessen des Referenzsystems YA
      LTR1(1,1)=0.
      LTR1(1,2)=0.
      LTR1(1,3)=0.
      LTR1(2,1)=0.
      LTR1(2,2)=0.
      LTR1(2,3)=0.
      LTR1(3,1)=0.
      LTR1(3,2)=0.
      LTR1(3,3)=0.
C>
      LRR1(1,1)=0.
      LRR1(1,2)=0.
      LRR1(1,3)=0.
      LRR1(2,1)=0.
      LRR1(2,2)=0.
      LRR1(2,3)=0.
      LRR1(3,1)=0.
      LRR1(3,2)=0.
      LRR1(3,3)=0.
C>
      AQTR1(1)=AX
      AQTR1(2)=AY
      AQTR1(3)=0.
C>
      AQRR1(1)=0.
      AQRR1(2)=0.
      AQRR1(3)=GAMMAOTOT
C>
      OR1(1)=0.
      OR1(2)=0.
      OR1(3)=GAMMAOT
C>
C>  Kinematische Groessen des Referenzsystems RO
      LTR2(1,1)=-SIN(THETA)
      LTR2(1,2)=0.
      LTR2(1,3)=0.
      LTR2(2,1)=BETA*COS(THETA)
      LTR2(2,2)=0.
      LTR2(2,3)=0.
      LTR2(3,1)=COS(THETA)
      LTR2(3,2)=0.
      LTR2(3,3)=0.
C>
      LRR2(1,1)=0.
      LRR2(1,2)=1.
      LRR2(1,3)=0.
      LRR2(2,1)=0.
      LRR2(2,2)=0.
      LRR2(2,3)=0.
      LRR2(3,1)=0.
      LRR2(3,2)=0.
      LRR2(3,3)=0.
C>
      AQTR2(1)=AX*COS(THETA)
      AQTR2(2)=AX*BETA*SIN(THETA)+AY
      AQTR2(3)=AX*SIN(THETA)-AY*BETA
C>
      AQRR2(1)=-GAMMAOTOT*SIN(THETA)

      AQRR2(2)=GAMMAOT*BETAP*COS(THETA)+
                  GAMMAOTOT*BETA*COS(THETA)
      AQRR2(3)=GAMMAOTOT*COS(THETA)
C>
      OR2(1)=-GAMMAOT*SIN(THETA)+BETAP
      OR2(2)=GAMMAOT*BETA*COS(THETA)
      OR2(3)=GAMMAOT*COS(THETA)
C>
C>  Kinematische Groessen des Referenzsystems PI
      LTR3(1,1)=0.
      LTR3(1,2)=0.
      LTR3(1,3)=0.
      LTR3(2,1)=BETA*COS(THETA)
      LTR3(2,2)=-HRC*COS(THETA)
      LTR3(2,3)=0.
      LTR3(3,1)=1.
      LTR3(3,2)=0.
      LTR3(3,3)=0.
C>
      LRR3(1,1)=0.
      LRR3(1,2)=COS(THETA)
      LRR3(1,3)=0.
      LRR3(2,1)=0.
      LRR3(2,2)=0.
      LRR3(2,3)=0.
      LRR3(3,1)=0.
      LRR3(3,2)=-SIN(THETA)
      LRR3(3,3)=0.
C>
      AQTR3(1)=-AY*BETA*SIN(THETA)+AX+2.*HRC*
      GAMMAOT*BETAP*COS(THETA)+ HRC*
      GAMMAOTOT*BETA*COS(THETA)
      AQTR3(2)=AX*BETA*SIN(THETA)+AY+
      HRC*GAMMAOT**2*BETA*COS(THETA)
      AQTR3(3)=-AY*BETA*COS(THETA)
C>
      AQRR3(1)=0.
      AQRR3(2)=GAMMAOT*BETAP*COS(THETA)+
      GAMMAOTOT*BETA*COS(THETA)
      AQRR3(3)=GAMMAOTOT
C>
      OR3(1)=BETAP*COS(THETA)
      OR3(2)=GAMMAOT*BETA*COS(THETA)
      OR3(3)=-BETAP*SIN(THETA)+GAMMAOT
C>
C>*************************************************
C>***                                             ***
C>***       A u s g a b e b l o c k  5 - E n d e   ***
C>***                                             ***
C>***                                             ***
C>*************************************************
C>***          A u s g a b e b l o c k  6          ***
C>***                                             ***
C>*** K i n e m a t i s c h e  G r o e s s e n  Koerper
C>***                                             ***
C>***                                             ***
C>*************************************************
```

```
C>  Kinematische Groessen des Starrkoerpers CAR
C>                                  im Inertialsystem
C>
      LT1(1,1)=0.
      LT1(1,2)=-HRC*BETA*SIN(THETA)*
      COS(THETA)*COS(GAMMA)+HRC*
      SIN(GAMMA)*COS(THETA)
      LT1(1,3)=0.
      LT1(2,1)=0.
      LT1(2,2)=-HRC*BETA*SIN(THETA)*
      SIN(GAMMA)*COS(THETA)-HRC*
      COS(THETA)*COS(GAMMA)
      LT1(2,3)=0.
      LT1(3,1)=1.
      LT1(3,2)=-HRC*BETA*COS(THETA)**2
      LT1(3,3)=0.
C>
      AQT1(1)=AX*COS(GAMMA)+2.*HRC*
      GAMMAOT*BETAP*COS(THETA)*
      COS(GAMMA)+HRC*GAMMAOTOT*BETA*
      COS(THETA)*COS(GAMMA)- AY*SIN(GAMMA)-
      HRC*GAMMAOT**2*BETA*SIN(GAMMA)*
      COS(THETA)
      AQT1(2)=AX*SIN(GAMMA)+2.*HRC*
      GAMMAOT*BETAP*SIN(GAMMA)*
      COS(THETA)+HRC*GAMMAOTOT*BETA*
      SIN(GAMMA)*COS(THETA)+AY*COS(GAMMA)+
      HRC*GAMMAOT**2*BETA*COS(THETA)*
      COS(GAMMA)
      AQT1(3)=0.
C>
      LR1(1,1)=0.
      LR1(1,2)=COS(THETA)*COS(GAMMA)
      LR1(1,3)=BETA*SIN(THETA)*COS(GAMMA)-
      SIN(GAMMA)
      LR1(2,1)=0.
      LR1(2,2)=SIN(GAMMA)*COS(THETA)
      LR1(2,3)=BETA*SIN(THETA)*SIN(GAMMA)+
      COS(GAMMA)
      LR1(3,1)=0.
      LR1(3,2)=-SIN(THETA)
      LR1(3,3)=BETA*COS(THETA)
C>
      AQR1(1)=-GAMMAOT*ALPHAP*COS(GAMMA)-
      GAMMAOT*BETAP*SIN(GAMMA)*COS(THETA)
      AQR1(2)=-GAMMAOT*ALPHAP*SIN(GAMMA)+
      GAMMAOT*BETAP*COS(THETA)*COS(GAMMA)
      AQR1(3)=GAMMAOTOT
C>
      O1(1)=BETAP*COS(THETA)*COS(GAMMA)-
      ALPHAP*SIN(GAMMA)
      O1(2)=BETAP*SIN(GAMMA)*COS(THETA)+
      ALPHAP*COS(GAMMA)
      O1(3)=-BETAP*SIN(THETA)+GAMMAOT
C>
C>*********************************************
C>***                                       ***
C>***       Ausgabeblock 6-Ende             ***
C>***                                       ***
C>***                                       ***
C>*********************************************
```

```
C>    MASSENGEOMETRISCHE DATEN
C>    ************************
C>
C>    Car with Pictch Motion
      KOSYNA:    CAR          Namen des
Koordinatensystems
C>
C>     Masse
      Mass =     MC
C>
C>     Traegheitstensor
      KOSYNA:    CAR     System fuer Angabe der
Traegheitstensors
C>
      I(1,1)= IX
      I(2,1)= 0
      I(2,2)= IY
      I(3,1)= 0
      I(3,2)= 0
      I(3,3)= IZ

C>*********************************************
C>***                                       ***
C>***          Ausgabeblock 7               ***
C>***                                       ***
C>***  Massengeometrische Groessen          ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>
C>  Massengeometrische Groessen fuer CAR
      MA1=MC
      I1(1,1)=IX*COS(GAMMA)**2+2.*IX*BETA*
      SIN(THETA)*SIN(GAMMA)*COS(GAMMA)-
      2.*IY*BETA*SIN(THETA)*SIN(GAMMA)*
      COS(GAMMA)+ IY*SIN(GAMMA)**2
      I1(2,1)=IX*SIN(GAMMA)*COS(GAMMA)+
      IX*BETA*SIN(THETA)*IN(GAMMA)**2-
      IX*BETA*SIN(THETA)*COS(GAMMA)**2-
      IY*BETA*SIN(THETA)*SIN(GAMMA)**2+
      IY*BETA*SIN(THETA)*COS(GAMMA)**2-
      IY*SIN(GAMMA)*COS(GAMMA)
      I1(2,2)=IX*SIN(GAMMA)**2- 2.*IX*BETA*
      SIN(THETA)*SIN(GAMMA)* COS(GAMMA)+
      2.*IY*BETA*SIN(THETA)*SIN(GAMMA)*
      COS(GAMMA)+IY*COS(GAMMA)**2
      I1(3,1)=-IX*ALPHA*COS(GAMMA)-IY*BETA*
      SIN(GAMMA)*COS(THETA)+ IZ*ALPHA*
      COS(GAMMA)+IZ*BETA*SIN(GAMMA)*
      COS(THETA)
      I1(3,2)=-IX*ALPHA*SIN(GAMMA)+IY*BETA*
      COS(THETA)*COS(GAMMA)+IZ*ALPHA*
      SIN(GAMMA)-IZ*BETA*COS(THETA)*
      COS(GAMMA)
      I1(3,3)=IZ

C>*********************************************
C>***                                       ***
C>***       Ausgabeblock 7-Ende             ***
C>***                                       ***
C>*********************************************
```

```
C>      EINGEPRAEGTE KRAEFTE
C>     **********************
C>
C>     Applied Force on CAR
C>     ---------------------------------------------
C>     Art der Kraft/des Moments
(GK,AK,AM,IK,IM,$END)
     FLEART:      AK
C>     System auf das die Kraft wirkt
     KOSYNA:      CAR        1.System (Wirkung positiv)
C>
C>     System, in dem die Kraft eingegeben wird
     KOSYNA:      CAR
C>
C>     Kraft/Moment eingeben
     FLE(1)=      0
     FLE(2)=      0
     FLE(3)=      FZC
C>
C>     Applied Moments on CAR
C>     ---------------------------------------------
C>     Art der Kraft/des Moments
(GK,AK,AM,IK,IM,$END)
     FLEART:      AM
C>
C>     System auf das das Moment wirkt
     KOSYNA:      CAR        1.System (Wirkung positiv)
C>
C>     System, in dem das Moment eingegeben wird
     KOSYNA:      CAR
C>
C>     Kraft/Moment eingeben
     FLE(1)=      LBETA
     FLE(2)=      LALPHA
     FLE(3)=      0
C>
C>*********************************************
C>
     FLEART:      $END
C
C>*********************************************
C>***                                       ***
C>***            Ausgabeblock 8             ***
C>***                                       ***
C>***           Kraefte/Momente             ***
C>***                                       ***
C>*********************************************
C>
C>   Kraefte/Momente auf CAR
C>              im Inertialsystem
     FE1(1)=FZC*ALPHA*COS(GAMMA)+FZC*
     BETA*SIN(GAMMA)*COS(THETA)
     FE1(2)=FZC*ALPHA*SIN(GAMMA)-FZC*BETA*
     COS(THETA)*COS(GAMMA)
     FE1(3)=FZC
C>
    LE1(1)=LALPHA*BETA*SIN(THETA)*
    COS(GAMMA)+LBETA*COS(GAMMA)-
    LALPHA*SIN(GAMMA)+LBETA*BETA*
    SIN(THETA)*SIN(GAMMA)
    LE1(2)=LALPHA*BETA*SIN(THETA)*
    SIN(GAMMA)+LBETA*SIN(GAMMA)+
    LALPHA*COS(GAMMA)-LBETA*BETA*
    SIN(THETA)*COS(GAMMA)
    LE1(3)=LALPHA*BETA*COS(THETA)-
    LBETA*ALPHA
C>
C>*********************************************
C>***                                       ***
C>***        Ausgabeblock 8-Ende            ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>*********************************************
C>***                                       ***
C>***           Ausgabeblock 9              ***
C>***                                       ***
C>***    NEWTON-EULER-Gleichungen           ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>
C>   Massenmatrix
     MQT1(1,1)=0.
     MQT1(1,2)=MC*HRC*SIN(GAMMA)*
     COS(THETA)
     MQT1(1,3)=0.
     MQT1(2,1)=0.
     MQT1(2,2)=-MC*HRC*COS(THETA)*
     COS(GAMMA)
     MQT1(2,3)=0.
     MQT1(3,1)=MC
     MQT1(3,2)=0.
     MQT1(3,3)=0.
     MQR1(1,1)=0.
     MQR1(1,2)=IX*COS(THETA)*COS(GAMMA)
     MQR1(1,3)=-IY*SIN(GAMMA)
     MQR1(2,1)=0.
     MQR1(2,2)=IX*SIN(GAMMA)*COS(THETA)
     MQR1(2,3)=IY*COS(GAMMA)
     MQR1(3,1)=0.
     MQR1(3,2)=-IZ*SIN(THETA)
     MQR1(3,3)=0.
C>
C>   Kreisel-, Zentrifugal- und Corioliskraefte
     KQT1(1)=MC*AX*COS(GAMMA)+2.*MC*
     HRC*GAMMAOT*BETAP*COS(THETA)*
     COS(GAMMA)+MC*HRC*GAMMAOTOT*
     BETA*COS(THETA)*COS(GAMMA)-
     MC*AY*SIN(GAMMA)-MC*HRC*GAMMAOT**2*
     BETA*SIN(GAMMA)*COS(THETA)
     KQT1(2)=MC*AX*SIN(GAMMA)+2.*MC*HRC*
     GAMMAOT*BETAP*SIN(GAMMA)*
     COS(THETA)+MC*HRC*GAMMAOTOT*BETA*
     SIN(GAMMA)*COS(THETA)+ MC*AY*
     COS(GAMMA)+MC*HRC*GAMMAOT**2*
     BETA*COS(THETA)*COS(GAMMA)
     KQT1(3)=0.
```

```
      KQR1(1)=-IX*GAMMAOTOT*ALPHA*
     COS(GAMMA)-IY*GAMMAOTOT*BETA*
     SIN(GAMMA)*COS(THETA)+IZ*GAMMAOTOT*
     ALPHA*COS(GAMMA)+ IZ*GAMMAOTOT*
     BETA*SIN(GAMMA)*COS(THETA)-
     IX*GAMMAOT*ALPHAP*COS(GAMMA)-
     IY*GAMMAOT*BETAP*SIN(GAMMA)*
     COS(THETA)+IX*GAMMAOT**2*
     ALPHA*SIN(GAMMA)-IY*GAMMAOT**2*BETA*
     COS(THETA)*COS(GAMMA)-IZ*
     GAMMAOT**2*ALPHA*SIN(GAMMA)+
     IZ*GAMMAOT**2*BETA*COS(THETA)*
     COS(GAMMA)-IX*GAMMAOT*BETAP*
     SIN(GAMMA)*COS(THETA)- IY*GAMMAOT*
     ALPHAP*COS(GAMMA)+IZ*GAMMAOT*BETAP*
     SIN(GAMMA)*COS(THETA)+IZ*GAMMAOT*
     ALPHAP*COS(GAMMA)
      KQR1(2)=-IX*GAMMAOTOT*ALPHA*
     SIN(GAMMA)+IY*GAMMAOTOT*BETA*
     COS(THETA)*COS(GAMMA)+IZ*GAMMAOTOT*
     ALPHA*SIN(GAMMA)- IZ*GAMMAOTOT*
     BETA*COS(THETA)*COS(GAMMA)-
     IX*GAMMAOT*ALPHAP*SIN(GAMMA)+
     IY*GAMMAOT*BETAP*COS(THETA)*
     COS(GAMMA)-IZ*GAMMAOT*BETAP*
     COS(THETA)*COS(GAMMA)+
     IZ*GAMMAOT*ALPHAP*SIN(GAMMA)-
     IX*GAMMAOT**2*ALPHA*COS(GAMMA)-
     IY*GAMMAOT**2*BETA*SIN(GAMMA)*
     COS(THETA)+ IZ*GAMMAOT**2*ALPHA*
     COS(GAMMA)+IZ*GAMMAOT**2*BETA*
     SIN(GAMMA)*COS(THETA)+IX*GAMMAOT*
     BETAP*COS(THETA)*COS(GAMMA)-
     IY*GAMMAOT*ALPHAP*SIN(GAMMA)
      KQR1(3)=IZ*GAMMAOTOT
C>
C>  Eingepraegte Kraefte/Momente
      QEQT1(1)=FZC*ALPHA*COS(GAMMA)+
     FZC*BETA*SIN(GAMMA)*COS(THETA)
      QEQT1(2)=FZC*ALPHA*SIN(GAMMA)-
     FZC*BETA*COS(THETA)*COS(GAMMA)
      QEQT1(3)=FZC
      QEQR1(1)=LALPHA*BETA*SIN(THETA)*
     COS(GAMMA)+LBETA*COS(GAMMA)-
     LALPHA*SIN(GAMMA)+LBETA*BETA*
     SIN(THETA)*SIN(GAMMA)
      QEQR1(2)=LALPHA*BETA*SIN(THETA)*
     SIN(GAMMA)+LBETA*SIN(GAMMA)+
     LALPHA*COS(GAMMA)-LBETA*BETA*
     SIN(THETA)*COS(GAMMA)
      QEQR1(3)=LALPHA*BETA*COS(THETA)-
     LBETA*ALPHA
C>
C>*********************************************
C>***                                       ***
C>***       Ausgabeblock 9-Ende             ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>*********************************************
C>***                                       ***
C>***          Ausgabeblock 10              ***
C>***                                       ***
C>***     Bewegungs-Gleichungen             ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>
C>  Massenmatrix
      M(1,1)=MC
      M(2,1)=0.
      M(2,2)=IZ*SIN(THETA)**2+MC*HRC**2*
     COS(THETA)**2+IX*COS(THETA)**2
      M(3,1)=0.
      M(3,2)=0.
      M(3,3)=IY
C>
C>  Verallgemeinerte Kreisel-, Zentrifugal- und
Corioliskraefte
      K(1)=0.
      K(2)=-IZ*GAMMAOTOT*SIN(THETA)-MC*HRC*
     AX*BETA*SIN(THETA)*COS(THETA)-MC*HRC*
     AY*COS(THETA)-MC*HRC**2*GAMMAOT**2*
     BETA*COS(THETA)**2-IX*GAMMAOTOT*
     ALPHA*COS(THETA)+IZ*GAMMAOTOT*
     ALPHA*COS(THETA)-IX*GAMMAOT*
     ALPHAP*COS(THETA)-IY*GAMMAOT**2*
     BETA*COS(THETA)**2+IZ*GAMMAOT**2*
     BETA*COS(THETA)**2-IY*GAMMAOT*
     ALPHAP*COS(THETA)+IZ*GAMMAOT*
     ALPHAP*COS(THETA)
      K(3)=IY*GAMMAOTOT*BETA*COS(THETA)+
     IY*GAMMAOT*BETAP*COS(THETA)-
     IZ*GAMMAOT*BETAP*COS(THETA)-
     IX*GAMMAOT**2*ALPHA+IZ*GAMMAOT**2*
     ALPHA+IX*GAMMAOT*BETAP*COS(THETA)
C>
C>  Verallgemeinerte eingepraegte Kraefte
      QE(1)=FZC
      QE(2)=LBETA*ALPHA*SIN(THETA)+LBETA*
     COS(THETA)
      QE(3)=LALPHA
C>
C>  Matrix der geschwindigkeitsproportionalen Kraefte
      P(1,1)=0.
      P(1,2)=0.
      P(1,3)=0.
      P(2,1)=0.
      P(2,2)=0.
      P(2,3)=-IX*GAMMAOT*COS(THETA)-
     IY*GAMMAOT*COS(THETA)+IZ*GAMMAOT*
     COS(THETA)
      P(3,1)=0.
      P(3,2)=IY*GAMMAOT*COS(THETA)-
     IZ*GAMMAOT*COS(THETA)+IX*GAMMAOT*
     COS(THETA)
      P(3,3)=0.
C>
```

```
C>   Matrix der gyroskopischen Kraefte
     G(1,1)=0.
     G(1,2)=0.
     G(1,3)=0.
     G(2,1)=0.
     G(2,2)=0.
     G(2,3)=-IX*GAMMAOT*COS(THETA)-
     IY*GAMMAOT*COS(THETA)+IZ*GAMMAOT*
     COS(THETA)
     G(3,1)=0.
     G(3,2)=IY*GAMMAOT*COS(THETA)-
     IZ*GAMMAOT*COS(THETA)+IX*GAMMAOT*
     COS(THETA)
     G(3,3)=0.
C>
C>   Matrix der Daempfungskraefte
     D(1,1)=0.
     D(1,2)=0.
     D(1,3)=0.
     D(2,1)=0.
     D(2,2)=0.
     D(2,3)=0.
     D(3,1)=0.
     D(3,2)=0.
     D(3,3)=0.
C>
C>   Matrix der lageproportionalen Kraefte
     Q(1,1)=0.
     Q(1,2)=0.
     Q(1,3)=0.
     Q(2,1)=0.
     Q(2,2)=-MC*HRC*AX*SIN(THETA)*COS(THETA)-
     MC*HRC**2*GAMMAOT**2*COS(THETA)**2-
     IY*GAMMAOT**2*COS(THETA)**2+
     IZ*GAMMAOT**2*COS(THETA)**2
     Q(2,3)=-IX*GAMMAOTOT*COS(THETA)+
     IZ*GAMMAOTOT*COS(THETA)-
     LBETA*SIN(THETA)
     Q(3,1)=0.
     Q(3,2)=IY*GAMMAOTOT*COS(THETA)
     Q(3,3)=-IX*GAMMAOT**2+IZ*GAMMAOT**2
C>
C>   Matrix der nichtkonservativen Lagekraefte
     N(1,1)=0.
     N(1,2)=0.
     N(1,3)=0.
     N(2,1)=0.
     N(2,2)=0.
     N(2,3)=-0.5*IX*GAMMAOTOT*COS(THETA)+
     0.5*IZ*GAMMAOTOT*COS(THETA)-0.5*LBETA*
     SIN(THETA)-0.5*IY*GAMMAOTOT*COS(THETA)
     N(3,1)=0.
     N(3,2)=0.5*IY*GAMMAOTOT*COS(THETA)+
     0.5*IX*GAMMAOTOT*COS(THETA)-
     0.5*IZ*GAMMAOTOT*COS(THETA)+
     0.5*LBETA*SIN(THETA)
     N(3,3)=0.
C>
C>   Matrix der konservativen Lagekraefte
     K(1,1)=0.
     K(1,2)=0.
     K(1,3)=0.
     K(2,1)=0.
     K(2,2)=-MC*HRC*AX*SIN(THETA)*
     COS(THETA)-MC*HRC**2*GAMMAOT**2*
     COS(THETA)**2-IY*GAMMAOT**2*
     COS(THETA)**2+IZ*GAMMAOT**2*
     COS(THETA)**2
     K(2,3)=-0.5*IX*GAMMAOTOT*COS(THETA)+
     0.5*IZ*GAMMAOTOT*COS(THETA)-
     0.5*LBETA*SIN(THETA)+0.5*IY*GAMMAOTOT*
     COS(THETA)
     K(3,1)=0.
     K(3,2)=0.5*IY*GAMMAOTOT*COS(THETA)-
     0.5*IX*GAMMAOTOT*COS(THETA)+
     0.5*IZ*GAMMAOTOT*COS(THETA)-
     0.5*LBETA*SIN(THETA)
     K(3,3)=-IX*GAMMAOT**2+IZ*GAMMAOT**2
C>
C>   Vektor der Steuer- und Stoerkraefte
     H(1)=FZC
     H(2)=IZ*GAMMAOTOT*SIN(THETA)+
     MC*HRC*AY*COS(THETA)+LBETA* COS(THETA)
     H(3)=LALPHA
C>
C>*********************************************
C>***                                       ***
C>***                                       ***
C>***      A u s g a b e b l o c k  10 - E n d e      ***
C>***                                       ***
C>***                                       ***
C>*********************************************
C>
C>   Liste der verwendeten Namen
  T          ZC         BETA       ALPHA
  ZCP        BETAP      ALPHAP     ZCPP
  BETAPP     ALPHAPP    XV         XVOT
  XVOTOT     YV         YVOT       YVOTOT
  GAMMA      GAMMAOT    GAMMAOTOT  AX
  AY         THETA      HRV        HRC
  MC         IX         IY         IZ
  FZC        LBETA      LALPHA
C>   Speicherplatzbelegung:
C>
C>                    benoetigt  vorhanden     frei
C>    Arbeitsfeld          3140       50000     46860
C>    Faktorfeld              4       20000     19996
C>    Variablenfeld          31        2000      1969
C>   Rechenzeit in Sekunden :    0.020
```

Appendix C

MATLAB S-Function for Simulation

Figure 3.7 shows the simulation model of the spatial car described in MATLAB/Simulink. The block '*car3dof_sfunc*' represents the linearized differential equations of the spatial car model according to Appendix B. The content of this S-function is shown below and consists of four parts:

- initialization and initial conditions,
- right-hand side of ODE,
- output definition,
- and termination.

```
% car3dof_sfunc.m
%
% Tuan-Anh. Nguyen   October, 10. 2005
%
% ======================================
%            Linearizad Equations of Motion of the
%                    SPATIAL CAR MODEL
% ======================================

function [sys,x0,str,ts] = car3dof_sfunc(t,x,u,flag,param)

switch flag,

% Initialization
%----------------
case 0,
   [sys,x0,str,ts]= mdlInitializeSizes;

% Derivatives
%---------------
case 1,
   [sys]=mdlDerivatives(t,x,u,param);

% Update
%----------
case 2,
   sys=mdlUpdate(t,x,u);

% Outputs
%-----------
case 3,
   sys=mdlOutputs(t,x,u);
   sys=[sys];

% Terminate
%-------------
case 9,
   sys=mdlTerminate(t,x,u);

% Unexpected flags
%----------------------
otherwise
   error(['Unhandled flag = ',num2str(flag)]);
end

% end sfuntmpl
%
%=====================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times
% for the S-function.
%=====================================
```

```
function [sys,x0,str,ts]=mdlInitializeSizes

   sizes = simsizes;
   sizes.NumContStates      = 6;
   sizes.NumDiscStates      = 0;
   sizes.NumOutputs         = 6;
   sizes.NumInputs          = 19;
   sizes.DirFeedthrough     = 0;
   sizes.NumSampleTimes = 1;
   sys = simsizes(sizes);

% initialize the initial conditions
   x0 = [0 0 0 0 0 0];

% str is always an empty matrix
   str = [];

% initialize the array of sample times
   ts  = [0 0];

% end Initial Conditions

%=======================================
% mdlDerivatives
% Return the derivatives for the continuous states.

%    LINEARIZED EQUATIONS OF MOTION OF
%                SPATIAL CAR MODEL
%           ------------------------------------
%           | xp = A*x + B*u + Bw*w  |          (1)
%           ------------------------------------
%=======================================

function [sys]=mdlDerivatives(t,x,u,param)

% reflection of the input to system parameter of NEWEUL

% generalized coordinates (Verallgemeine Koordinaten)
%
   ZC       = u(1);
   BETA     = u(2);
   ALPHA = u(3);

% generalized velocities (Verallgemeine Geschwindigkeit)
%
   ZCP      = u(4);
   BETAP    = u(5);
   ALPHAP = u(6);

% parameters of car motion
%
   XVOTOT       = u(7);
   YVOTOT       = u(8);
   GAMMAOTOT= u(9) ;

   XVOT         = u(10);
   YVOT         = u(11);
   GAMMAOT  = u(12);

   XV     = u(13);
   YV     = u(14);
   GAMMA   = u(15);

% control forces

   u1 = u(16);
   u2 = u(17);
   u3 = u(18);
   u4 = u(19);

   u = [u1; u2; u3; u4];

% car accelerations

   AX = XVOTOT*cos(GAMMA) + …
        YVOTOT*sin(GAMMA);
   AY =-XVOTOT*sin(GAMMA) + …
        YVOTOT*cos(GAMMA);

% vector of disturbances

   w = [AY; GAMMAOTOT];

% end of the reflection
%
% ---------------------------------------------
%    PARAMETERS OF CAR MODEL
% ---------------------------------------------

% car parameters
% ------------------
   THETA = 15/180*pi;

   HRC  = 0.2;   % distance of the sprung mass c.g. from
                 % the roll axes
   HRV  = 0.3;   % height of the roll axes
   MC   = 1460;  % sprung mass
   IX   = 460;   % roll moment of inertia of the sprung
                 % mass
   IY   = 2460;  % pitch moment of inertia of the sprung
                 % mass
   IZ   = 1900;  % yaw moment of inertia of the sprung
                 % mass

   bf   = 1290;  % front suspension damping rate
   br   = 1620;  % rear suspension damping rate
   kf = 19960;   % front suspension stiffness
   kr = 17500;   % rear suspension stiffness
   rf = 19200;   % modified front anti-roll bar stiffness
   rr = 9600;    % modified rear anti-roll bar stiffness

   tf = 0.761;   % half the distance between the front
                 % wheels
   tr = 0.755;   % half the distance between the rear
                 % wheels
   lf = 1.011;   % distance between the c.g. and the front
                 % axle
   lr = 1.803;   % distance between the c.g. and the rear
                 %  axle
%
% suspension stiffness and damping matrices
% ------------------------------------------------
```

```
   Kss = diag([kf,kf,kr,kr]);
   Ksr = [rf,-rf,0,0; -rf,rf,0,0,; 0,0,rr,-rr; 0,0,-rr,rr];
   Bs  = diag([bf bf br br]);
   G   = [1,1,1,1; tf,-tf,tr,-tr; -lf,-lf,lr,lr];
   Gs  = G; Gs(2,:)= G(2,:)*cos(THETA);

% end of initialization
%
% ------------------------------------------------------------------
% system matrices definition
%
% mass matrix

    M(1,1)=MC;
    M(2,1)=0.;
    M(2,2)=IZ*sin(THETA)^2+MC*HRC^2*…
           cos(THETA)^2+IX*cos(THETA)^2;

    M(3,1)=0.;
    M(3,2)=0.;
    M(3,3)=IY;

% matrix of velocity dependent forces

    P(1,1)=0.;
    P(1,2)=0.;
    P(1,3)=0.;

    P(2,1)=0.;
    P(2,2)=0.;
    P(2,3)=(-IX-IY+IZ)*GAMMAOT*cos(THETA);

    P(3,1)=0.;
    P(3,2)=(IY-IZ+IX)*GAMMAOT*cos(THETA);
    P(3,3)=0.;

% matrix of position dependent forces

    Q(1,1)=0.;
    Q(1,2)=0.;
    Q(1,3)=0.;

    Q(2,1)=0.;
    Q(2,2)=-MC*HRC*sin(THETA)*cos(THETA)*AX-...
           (MC*HRC^2+IY-IZ)*GAMMAOT^2*…
            cos(THETA)^2;
    Q(2,3)=(IZ-IX)*GAMMAOTOT*cos(THETA);

    Q(3,1) =0.;
    Q(3,2) = IY*GAMMAOTOT*cos(THETA);
    Q(3,3) = (IZ-IX)*GAMMAOT^2;

% exciting vector

    H(1) = 0;
    H(2) = MC*HRC*cos(THETA)*AY + …
           IZ*GAMMAOTOT*sin(THETA);
    H(3) = 0;

    H     = [0; MC*HRC*cos(THETA)*AY + …
           IZ*GAMMAOTOT*sin(THETA); 0];

    Hw   = [0, 0; MC*HRC*cos(THETA),…
            IZ*GAMMAOTOT*sin(THETA); 0, 0];

%  state matrices
%
   A = [zeros(3,3),eye(3);
        -inv(M)*(Q + Gs*(Kss+Ksr)*Gs'),...
        -inv(M)*(P + Gs*Bs*Gs')];
   B = [zeros(3,4); inv(M)*Gs];
   Bw = [zeros(3,2); inv(M)*Hw];

%measurement matrices
%
   gbeT= [0 1 0];
   C  = [Gs', zeros(4,3); gbeT*A(4:6,:)];
   D  = [zeros(4,4); gbeT*inv(M)*Gs];
   Dw = [zeros(4,2); gbeT*inv(M)*Hw];

% =====================================
% Right-hand side of Eq.(1)

   yp = A*x + B*u + Bw*w;

   sys(1:6) = [yp];

% end mdlDerivatives
%
%=====================================
% mdlUpdate
% Handle discrete state updates, sample time hits, and
% major time step requirements.
%=====================================
%
function sys=mdlUpdate(t,x,u)

% no discrete state updating necessary!
   sys = [];

% end mdlUpdate
%
%=====================================
% mdlOutputs
% Return the block outputs.
%=====================================
%
function sys=mdlOutputs(t,x,u)

   sys(1:6) = x(1:6);

% end mdlOutputs
%
% =====================================
% mdlTerminate
% Perform any end of simulation tasks.
%=====================================
%
function sys=mdlTerminate(t,x,u)

   sys = [x];

% end mdlTerminate
```

Appendix D

MATLAB.m Files for Optimization

This appendix describes the MATLAB.m files used to solve the multi-criterion optimization (MCO) problems defined for the spatial car model with passive and active suspensions introduced in Chapter 5 and the path generation problem for the ISO 3882-2 double-lane-change manuever presented in Chapter 6.

D.1 Suspension Optimization

In Section 5.5 the MCO problem defining the optimal parameters for both passive and active suspensions of the spatial car model is solved based on the compromise method by using the MATLAB optimization function '*fmincon*'. Ordinarily, the function '*fmincon*' requires two MATLAB.m files: one defines the objective function and the other one defines the nonlinear constraints. The corresponding MATLAB.m files named 'pass_act_object.m' and 'pass_act_const.m' are shown below.

```
% cm_pass_act.m
%
% Tuan-Anh Nguyen, November 2005
% ======================================
%
%      Compromise Method with LQR Algorithm
%          for MCO problem finding optimial
%         passive-active suspension parameters
%
% ======================================
```

```
function [PM]=cm_pass_act(f1min,f12,N)

% f1min    inidividual minimum f1
% f12      value f1 at f2min
%          (f1min & f12 are provided
%          from individual optimization)
% N        number of calculation points
% call program e.g.
% [PM] = cm_pass_act(0.3918,0.5243,19)
%-------------------------------------------------
```

```
global   vc  kappa  tsim  Gs  M  P  Q  Hw  f1  f3

tsim = 2.5;        % simulation time
vc=30/3.6;         % car speed
kappa = 1/10;      % curvature

%-------------------------------------
% car parameters
   THETA = 15/180*pi; % slope angle of roll axis
   HRC = 0.2;   % distance of the sprung mass c.g. from
                % the roll axes
   HRV= 0.3;    % height of the roll axes
   MC  = 1460; % sprung mass
   IX    = 460;    % roll moment of inertia of the
                % sprung mass
   IY    = 2460;  % pitch moment of inertia of the
                % sprung mass
   IZ    = 1900;  % yaw moment of inertia of the
                % sprung mass
   tf = 0.761;     % half the distance between the front wheel
   tr = 0.755;     % half the distance between the rear wheel
   lf = 1.011;     % distance between the c.g. and the front axle
   lr = 1.803;     % distance between the c.g. and the rear axle

   G   = [1,1,1,1; tf,-tf,tr,-tr; -lf,-lf,lr,lr];
   Gs  = G; Gs(2,:)= G(2,:)*cos(THETA);

%-------------------------------------------------
GAMMAOT = vc*kappa; GAMMAOTOT = 0; AX=0;

% mass matrix (massenmatrix)
   M(1,1)=MC;
   M(2,1)=0.;
   M(2,2)=IZ*sin(THETA)^2+MC*HRC^2*...
            cos(THETA)^2+IX*cos(THETA)^2;
   M(3,1)=0.;
   M(3,2)=0.;
   M(3,3)=IY;

% matrix of velocity dependent forces
   P(1,1)=0.;
   P(1,2)=0.;
   P(1,3)=0.;
   P(2,1)=0.;
   P(2,2)=0.;
   P(2,3)=(-IX-IY+IZ)*GAMMAOT*cos(THETA);
   P(3,1)=0.;
   P(3,2)=(IY-IZ+IX)*GAMMAOT*cos(THETA);
   P(3,3)=0.;

% matrix of position dependent forces
   Q(1,1)=0.;
   Q(1,2)=0.;
   Q(1,3)=0.;
   Q(2,1)=0.;
   Q(2,2)=-MC*HRC*sin(THETA)*cos(THETA)*AX-...
            (MC*HRC^2+IY-IZ)*GAMMAOT^2*…
            cos(THETA)^2;
   Q(2,3)=(IZ-IX)*GAMMAOTOT*cos(THETA);
   Q(3,1)=0.;
   Q(3,2)= IY*GAMMAOTOT*cos(THETA);
   Q(3,3) = (IZ-IX)*GAMMAOT^2;

% disturbance vertor
   Hw   = [0; MC*HRC*cos(THETA); 0];

%-----------------------------------------------------------------
%       Compromise method with LQR algorithm
% ----------------------------------------------------------------

% standard options for optimization with fmincon
OPTIONS = optimset('fmincon');
OPTIONS = optimset(OPTIONS,'TolX',1e-6,…
                    'TolFun',1e-3, 'TolCon',1e-3,...
                    'DiffMinChange',1e-9,…
                    'DiffMaxChange',1e-3,...
                    'LargeScale','off',…
                    'MaxFunEvals',1e3);

%initail design
p0  = [1500, 1500, 20000, 20000, 20000, 10000,…
        2e7     1e3     1e-2];
   % p(1):   front suspension damping rate
   % p(2):   rear suspension damping rate
   % p(3):   front suspension stiffness
   % p(4):   rear suspension stiffness
   % p(5):   front anti-roll bar stiffness
   % p(6):   rear anti-roll bar stiffness
   % p(7):   w1 weighting factor on f1
   % p(8):   w2 weighting factor on f2
   % p(9):   w3 weighting factor on f3

%lower and upper bounds
plb  = [1000, 1000, 15000, 15000, 15000, 5000,…
         0      0     1e-6];
pub = [2000, 2000, 25000, 25000, 25000, 15000,…
         1e8    1e6    1e-2];

%optimization algorithm

global  mui_f    % changing step of constraint bounds

% step changing for constraint bounds
delta_f1 = (f12-f1min)/N;

i = 1;
% set changing constraint bounds
for  mui_f = f1min + N*delta_f1 :-delta_f1 : f1min;

Z = clock;        % calculation time

[p,f,exitflag,output] = fmincon(@pass_act_object,…
                        p0,[],[],[],[],plb,pub,...
                        @pass_act_const,OPTIONS,…
                        f1min,f12,N)
if exitflag == -1
   break            % stop program if no solution is found
else
```

```
% outputs
   MCO_p(i,:)= p;
   MCO_f2(i) = f;
   MCO_f3(i) = f3;
   MCO_f1(i) = f1;
   PM(i,:)=[MCO_f1(i)',MCO_f2(i)',…
            MCO_f3(i)',MCO_p(i,:)];
   figure(1);
   xlabel('ride safety f_1 [cm]');
   ylabel('ride comfort f_2 [deg/s^2]');
   title('Criterion Space f_1 - f_2');
   plot(PM(:,1),PM(:,2), 'b.'); hold on;
   plot(PM(i,1),PM(i,2), 'ro'); hold on; grid;

% new start point
p0=p;

i = i+1;
end
end

disp(['Calculation time t = ' num2str(etime(clock,Z))...
     ' s  = ' num2str(etime(clock,Z)/60) ' min. '])

%=======================================
%
%            Objective function definition
%           for  cm_pass_act(f1min,f12,N)
%
%=======================================

function [f] = pass_act_object(p,f1min,f12,N)

disp(['--------------------------------'])
disp(['p =',' ',num2str(p)])

global  vc  kappa  tsim  Gs  M  P  Q  Hw  f1  f3
global  h  g  mui_f

% design variables
% --------------------
   bf = p(1);
   br = p(2);
   kf = p(3);
   kr = p(4);
   rf = p(5);
   rr = p(6);
   Bs  = diag([bf bf br br]);
   Kss = diag([kf,kf,kr,kr]);
   Ksr = [rf,-rf,0,0; -rf,rf,0,0,; 0,0,rr,-rr; 0,0,-rr,rr];

% system matrices
   A   = [zeros(3,3),eye(3);
          -inv(M)*(Q + Gs*(Kss+Ksr)*Gs'),...
          -inv(M)*(P + Gs*Bs*Gs')];
   B    = [zeros(3,4); inv(M)*Gs];
   Bw = [zeros(3,1); inv(M)*Hw];

% measurement matrices
   gbeT= [0 1 0];
   C     = [Gs', zeros(4,3); gbeT*A(4:6,:)];
   D     = [zeros(4,4); gbeT*inv(M)*Gs];
   Dw  = [zeros(4,1); gbeT*inv(M)*Hw];

% LQR algorithm
%  ------------------
   SYS = ss(A,B,C,D);

   % weighting matrices
   Qy = diag([p(7), p(7), p(7), p(7), p(8)]);
   Ry = diag([p(9), p(9), p(9), p(9)]);

   % optimal gain matrices K1& K2
   [Kx,Px,E]=lqry(SYS,Qy,Ry);
          R     =D'*Qy*D+Ry;
          Nuw=Dw'*Qy*D;
          Nxw=C'*Qy*Dw;
   Kw=(R)\[Nuw'+B'*inv(A'+Kx'*B')*…
         (Kx'*Nuw'-Nxw-Px*Bw)];

% simulation options
my_opt = simset('Initialstep', 0.01,…
                      'SrcWorkspace', 'current');

% call simulation
sim('cm_pass_act_model',tsim, my_opt);

% criteria definition
f1 = sqrt(sum(J_zs)/length(J_zs));            % ride safety
f2 = sqrt(sum(J_bepp)/length(J_bepp));    % ride comfort
f3 = sqrt(sum(J_u)/length(J_u));              % control effort

% objective function
f = f2;
   disp(['=== ','f1 = ',num2str(f1),' === ',...
         'f2 = ',num2str(f2),' === ','f3 = ',num2str(f3)])
   disp(['mui_f =',' ',num2str(mui_f)])

% nonlinear constraints
h(1) = f1 - mui_f;
h(2) = max(max(abs(ui)))-500;

g=[]; % no equality constraint

%=======================================
%
%             Nonlinear constraints definition
%             for  cm_pass_act(f1min,f12,N)
%
%=======================================

function [h,g] = pass_act_const(p,f1min,f12,N)

global h  g      % see pass_act_object(p,f1min,f12,N)
                 % for constraint computation
```

D.2 Path Generation for Double-Lane-Change Manuever

The following MATLAB.m file is used to solve the path generation problem for the ISO-3882-2 double-lane-change maneuver presented in Section 6.3.2. The entrance speed of the car is assumed as v = 60 *km/h* reducing steadily with a rate *rate_v* = – 0.5 m/s^2. The path is defined based on 11 parameters: the time points of changing curvature $t_1 - t_6$, the rates of curvature change (assumed to be the same) and the curvatures $\kappa_1 - \kappa_4$. The constraints for the problem are defined based on the positions of the four wheels during simulation and the wheel track boundaries given in Table 6.3.

```
% Elk_test_optim.m
%
% Tuan-Anh Nguyen, Apr. 2006
%=========================================
%
%  Path generation for ISO Doubl_Lane_Change Manuever
%                          (Elk Test)
%
% =========================================

function [PE]= Elk_test_optim

% call program
% [PE]=Elk_test_optim
%--------------------------

% simulation parameters
% -------------------------
v        = 60/3.6;    % entrance car speed
rate_v = -0.5;        % reduction rate of car speed
tsim    = 6;          % simulation time

global  tsim  v  rate_v  poff  pskal

% initial design
p0 = [ 1.05,    1.68,      2.45,      2.60,      3.20,      4.00,...
         0.3000,...
         0.0300,       -0.0300,      -0.0300,       0.0300]';

% bounds on design variables
kap_max = 9.81/v^2;    % from sideway slipping condition

plb = [ 0.5      1.0      1.5      2.0      3.0      3.5,...
         0.1000,...
         0.01     -kap_max     -kap_max     0.01]';
pub = [1.5       2.0      2.5      3.0      4.0      4.5,...
         0.3145,...
         kap_max     -0.01     -0.01      kap_max]';

% scalarization for optimizing on [0 1]
pskal= 1./(pub-plb);              % scalarization factor
poff  = plb;                      % offset
x_lb  = pskal.*(plb-poff);        % lower bound
x_ub = pskal.*(pub-poff);         % upper bound
x0     = pskal.*(p0-poff);        % start point

% standard options for optimization with fminimax
OPTIONS = optimset('fminimax');
OPTIONS = optimset(OPTIONS,'TolX',1e-3,…
                   'TolFun',1e-3,'TolCon',1e-3,...
                   'DiffMinChange',1e-9,…
                   'DiffMaxChange',1e-6,...
                   'LargeScale','off',…
                   'MaxFunEvals',1e3,'MinAbsMax',4);
Z = clock;

[x,f,exitflag] = fminimax(@Elk_test_object,x0,…
                   [],[],[],[],x_lb,x_ub,...
                   @Elk_test_const,OPTIONS)

% set outputs
p=poff+x./pskal;      % scalarization turnback
PE=p;

disp(['Calculation time t = ' num2str(etime(clock,Z))...
      ' s  = ' num2str(etime(clock,Z)/60) ' min. '])

%=====================================
%
%                Objective function definition
%                     for Elk_test_optim
%    minimize maximum kappa <=> minimize max ay
%
%=====================================

function [f] = Elk_test_object(x)

global   tsim  v  rate_v poff  pskal
global   h g

p=poff+x./pskal;
p=p';
```

```
disp(['-------------------------------'])
disp(['t =',' ',num2str(p(1:6))])
disp(['])
disp(['rate =',' ',num2str(p(7))])
disp(['])
disp(['kappa =',' ',num2str(p(8:11))])

% new design variables for simulation
   t_1=p(1);          t_2=p(2);          t_3=p(3);
   t_4=p(4);          t_5=p(5);          t_6=p(6);
   rate_1=p(7);       rate_2=-p(7);      rate_3=p(7);
   rate_4=-p(7);      rate_5=p(7);       rate_6=-p(7);
   kappa_1=p(8);      kappa_2=p(9);
   kappa_3=p(10);     kappa_4=p(11);

% simulation option
my_opt = simset('Initialstep', 0.01,...
                'SrcWorkspace', 'current');
% call simulation
[tS,yS]= sim('Elk_test_opt', tsim, my_opt);

% objective vector for minmax problem
f(1) = abs(kappa_1);
f(2) = abs(kappa_2);
f(3) = abs(kappa_3);
f(4) = abs(kappa_4);

% description of the position of test cones for Elch Test
%-----------------------------------------------------------------
%longitudinal position of cones
   xc1=[10:12/4:22];
   xc2=[(22+13.5):11/4:(35.5+11)];
   xc3=[(46.5+12.5):12/4:(59+12)];
   xcones=[xc1,xc2,xc3];

 % lateral position of upper and lower cones
   ycu1=[1.25*ones(1,length(xc1))];
   ycu2=[5.3*ones(1,length(xc2))];
   ycu3=[1.5*ones(1,length(xc3))];

   ycl1=[-1.25*ones(1,length(xc1))];
   ycl2=[2.25*ones(1,length(xc2))];
   ycl3=[-1.5*ones(1,length(xc3))];

   yc1 =[ycu1;ycl1];
   yc2 =[ycu2;ycl2];
   yc3 =[ycu3;ycl3];
   ycones=[yc1,yc2,yc3];

% lateral position of wheels
% ----------------------------
for i=1:1:4,
   x=xy_wheel(:,i);          % longitudinal position of
                             % wheel ith
   y=xy_wheel(:,i+4);        % lateral position of
                             % wheel ith
% section 1 (cone section)
[a,k0]=min(abs(xvS-10));     % start of section
[a,k1]=min(abs(xvS-22));     % end of section
   yw1=y(k0:k1);
% section 2 (non-cone section)
[a,k2]=min(abs(xvS-35.5));
% section 3 (cone section)
[a,k3]=min(abs(xvS-46.5));
   yw3=y(k2+1:k3);
% section 4 (non-cone section)
[a,k4]=min(abs(xvS-59));
% section 5 (cone section)
   yw5=y(k4+1:end);

%  nonlinear inequality constraints
%--------------------------------------
if  mod(i,2)==1        % constraints for left wheels
% section 1
   h(3*i-2)=max(yw1)-(1.25-0.3);
% cone section 3
   h(3*i-1)=max(yw3)-(5.30-0.3);
% cone section 5
   h(3*i)=max(yw5)-(1.5-0.3);

elseif  mod(i,2)==0    % constraints for right wheels
% section 1
   h(3*i-2)=-min(yw1)+(-1.25+0.3);
% section 3
   h(3*i-1)=-min(yw3)+(2.25+0.3);
% section 5
   h(3*i)=-min(yw5)+(-1.5+0.3);

end                    % if
end                    % for

% nonlinear equality constraints
%-------------------------------------
g = [];   % no equality constraint

figure(1)
   title('Car Path'); xlabel('x_I [m]'); ylabel('y_I [m]');
   plot (xvS,yvS,'b','LineWidth',1.5);hold on;
   plot (x3,y3,'g:');hold on;
   plot (x4,y4,'g:');hold on;
   plot (x1,y1,'b:');hold on;
   plot (x2,y2,'b:');hold on;
   plot (xcones,ycones,'r.','MarkerSize',5); hold off; grid;


%=====================================
%
%                  Constraint definitions
%                    for Elk_test_optim
%
% =====================================

function [h,g] = Elk_test_const(x)

global h  g            % see Elk_test_object(x)
                       % for constraint definition
```