

Optimization of multibody systems using approximation concepts

Citation for published version (APA):

Etman, L. F. P. (1997). *Optimization of multibody systems using approximation concepts*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR496429>

DOI:

[10.6100/IR496429](https://doi.org/10.6100/IR496429)

Document status and date:

Published: 01/01/1997

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

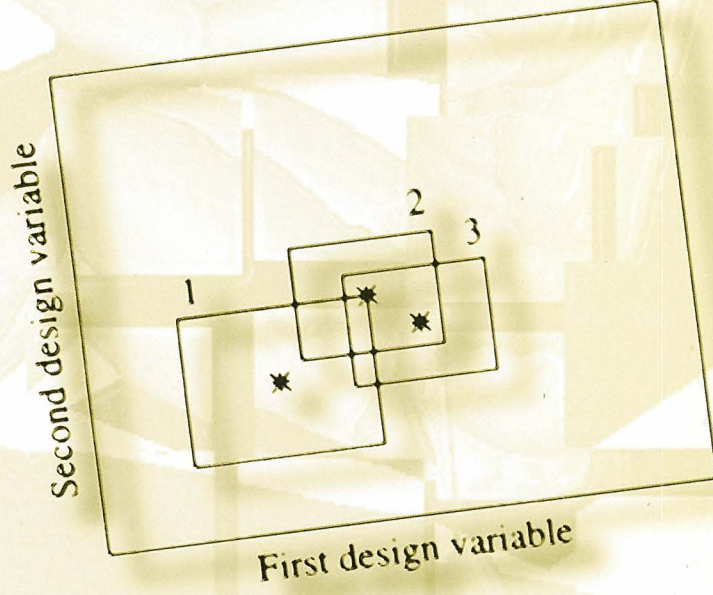
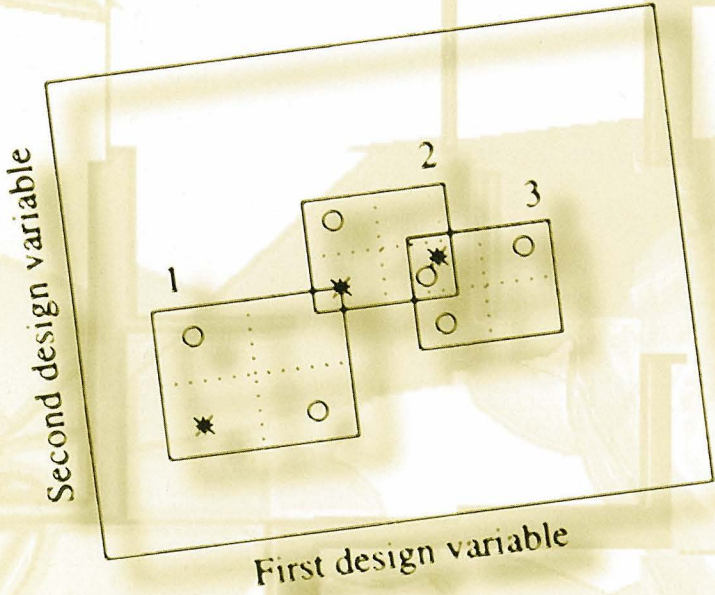
Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Optimization of Multibody Systems using Approximation Concepts



Pascal Etman

Optimization of Multibody Systems using Approximation Concepts

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Optimization of multibody systems using approximation concepts /
Lodewijk Franciscus Pascal Etman. - Eindhoven : Technische Universiteit
Eindhoven, 1997. - XVIII, 140 p. - With ref. - With summary in Dutch
Proefschrift. - ISBN 90-386-0520-X

NUGI 834

Trefwoorden: dynamica ; meerdelige systemen / numerieke optimalisering /
constructie-optimalisering / benaderingsconcepten / voertuigbotsingen /
schokdemping

Subject headings: multibody systems / numerical optimization /
optimum design / approximation concepts / crashworthiness design /
vehicle suspension

Druk: Universiteitsdrukkerij TU Eindhoven

Support by the BRITE/EURAM project OPTIM is gratefully acknowledged.

Optimization of Multibody Systems using Approximation Concepts

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof.dr. M. Rem,
voor een commissie aangewezen door het College van Dekanen
in het openbaar te verdedigen op
woensdag 18 juni 1997 om 16.00 uur

door

LODEWIJK FRANCISCUS PASCAL ETMAN

geboren te Hoorn

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. D.H. van Campen

prof.dr.ir. C. Fleury

en de copromotor:

dr.ir. A.J.G. Schoofs

Weisheit ist jeden Umweg wert

From: Hans Bemann,
Stein und Flöte

Voor Frans en Wies

Contents

Summary	xi
Notation	xiii
1 Introduction	1
1.1 Analysis and Optimization	1
1.2 Structural Optimization	2
1.3 Optimization of Multibody Systems	3
1.4 Objective and Outline	4
1.5 A Guideline for the Reader	5
2 Optimization Problem Formulation	7
2.1 Introduction	7
2.2 Optimization Problem	9
2.3 Objective Function and Constraints	10
2.4 Dealing with Time Dependent Constraints	12
2.5 Examples	14
2.5.1 Impact-Absorber	14
2.5.2 Slider-Crank Mechanism	17
2.6 Discussion	19
3 Sequential Approximate Optimization	21
3.1 Introduction	21
3.2 Approximation Concepts	22
3.3 Intermediate Variables and Responses	24
3.4 Approximate Optimization Problem	25
3.5 Design Optimization Tool for Multibody Systems	27
3.5.1 Approach	27
3.5.2 Constraint Deletion	28

- 3.5.3 Approximation 29
- 3.5.4 Move Limit Strategy 31
- 3.5.5 Aspects of Implementation 33
- 3.6 Multi-Point Approximations for Noisy Behavior 34
 - 3.6.1 Approach 34
 - 3.6.2 Approximation 35
 - 3.6.3 Move Limit Strategy 36
- 3.7 Examples 37
 - 3.7.1 Impact-Absorber 37
 - 3.7.2 Slider-Crank Mechanism 42
- 3.8 Conclusion and Discussion 44

- 4 Design Sensitivity Analysis 45**
 - 4.1 Introduction 45
 - 4.2 Finite Differences 48
 - 4.3 Kinematically Driven Multibody Systems 49
 - 4.3.1 Algebraic Equations of Motion 49
 - 4.3.2 Algebraic Sensitivity Equations 51
 - 4.3.3 Discussion 54
 - 4.4 Multibody Systems described by Ordinary Differential Equations 54
 - 4.4.1 Ordinary Differential Equations of Motion 54
 - 4.4.2 First-Order Differential Sensitivity Equations 55
 - 4.4.3 Discussion 58
 - 4.5 Multibody Systems described by Differential-Algebraic Equations 58
 - 4.5.1 Differential-Algebraic Equations of Motion 58
 - 4.5.2 Second-Order Differential-Algebraic Sensitivity Equations 60
 - 4.5.3 First-Order Ordinary Differential Sensitivity Equations 63
 - 4.5.4 Discussion 64
 - 4.6 Aspects of Practical Implementation 65
 - 4.7 Examples 67
 - 4.7.1 Impact-Absorber 67
 - 4.7.2 Slider-Crank Mechanism 71
 - 4.8 Conclusion and Discussion 74

- 5 Stress Constrained Design of a Four-Bar Mechanism 77**
 - 5.1 Introduction 77
 - 5.2 Design Optimization 78
 - 5.3 Influence of Dynamics 80
 - 5.4 Effect of Intermediate Variables and Responses 83

5.5	Conclusion and Discussion	83
6	Optimum Crashworthiness Design of a Vehicle Restraint System	87
6.1	Introduction	87
6.2	Crashworthiness Optimization Problem	88
6.3	Optimization using MADYMO	89
6.4	Restraint System for Frontal Impact	91
6.5	Optimization Results	93
6.6	Conclusion and Discussion	95
7	Design of a Stroke Dependent Damper for the Front Axle Suspension of a Truck	97
7.1	Introduction	97
7.2	Design Problem	99
7.3	Quarter Car Model	101
	7.3.1 Problem Description	101
	7.3.2 Optimization Results	104
7.4	Full-Scale Model	107
	7.4.1 Problem Description	107
	7.4.2 Optimization Results	109
7.5	Conclusion and Discussion	112
8	Conclusions and Recommendations	115
8.1	Problem Description	115
8.2	How Approximation Concepts can help	116
8.3	Approximate Optimization Tool	117
8.4	Intermediate Variables and Responses	118
8.5	Design Sensitivity Analysis	118
8.6	Further Recommendations	120
A	Move Limit Strategy	121
	Bibliography	125
	Samenvatting	133
	Curriculum Vitae	137
	Dankwoord	139

Summary

Multibody systems are mechanical models of interconnected rigid or flexible bodies. They are commonly employed in mechanical kinematics and dynamics of, for example, vehicles, mechanisms and robots. Multibody system analysis software is available to automatically build and solve the governing equations of motion for a variety of multibody systems. This enables the engineer to analyze and improve the multibody system design early in the design stage.

Systematic design improvements are rather difficult if the multibody system has multiple design parameters and several conflicting design criteria. Here, numerical optimization methods can aid. To this end, the multibody system analysis software has to be extended, and coupled with a suitable optimization strategy. The question is then how this coupling should be established to obtain a general purpose multibody design tool with a reliable and efficient design optimization process.

Mathematically formulated, the optimization problem is to find a set of design variable values that will minimize an objective function subject to constraints. Several mathematical programming algorithms are available that can solve such a problem. However, the optimization of multibody systems is hindered by the transient responses that follow from the numerical multibody analysis. The accompanying computational cost may be high, which limits the number of multibody analyses that can be carried out during the optimization. Another difficulty is that for a successful optimization the designer should not be completely left out of the optimization loop. User-interaction is required such that engineering experience and computer power can be combined to solve the optimization problem.

The approach adopted in this thesis is to couple multibody analysis and optimization algorithm by means of approximation concepts. Such an interface is computationally more convenient than a direct coupling, and enables the design engineer to influence the optimization process. This is demonstrated by means of design optimization software that has been especially developed for multibody systems, starting from single-point local and multi-point mid-range approximations. The software generates approximations of objective function and constraints in

a search subregion of the design space, and solves the approximate optimization problem separately from the multibody analysis. Afterwards, the search subregion is moved towards the calculated optimum design, which means that a sequence of approximate optimization cycles follows to reach the final optimum design.

The local approximation concept uses function values and derivatives with respect to the design variables in a single point of the design space. It is especially suitable for smooth functional behavior. The local concept relies on accurate sensitivities which may not be obtained by means of a finite difference sensitivity analysis. A computationally better and more efficient approach is to assemble and solve the sensitivity equations corresponding to the multibody equations of motion. Within this context either the direct differentiation or the adjoint variable method can be used. The direct method appears to be most suited to calculate the sensitivities for sequential approximate optimization using time discretization.

The mid-range concept builds approximations using numerical multibody analyses of multiple design points spread over the search subregion. It does not use sensitivities and is appropriate whenever a band of noise is present on the global responses. In that case a reasonable estimation of the magnitude of the noise is important for a good convergence.

The approximate optimization approach is illustrated for two analytical examples and three larger multibody design examples. For the latter design applications the optimization tool has been coupled with external multibody analysis codes. Both smooth and non-smooth design optimization problems have been successfully solved. The examples show that the transient multibody responses can be dealt with by means of time discretization, provided that a constraint deletion is applied. This constraint deletion retains only the most important time point constraints to obtain an approximate optimization problem of manageable proportions. Furthermore, the examples prove that suitable intermediate design variables and intermediate response quantities can be selected that improve the approximations and yield a robust and fast optimization process.

The main conclusion of this thesis is that approximation concepts can be effectively used for the design optimization of multibody systems. They provide an interface in between multibody analysis and optimization that allows user-interaction. This gives the opportunity to develop an interactive computer aided design tool for multibody systems.

Notation

Convention

a, A, α	scalar
\mathbf{a}	column
\mathbf{A}	matrix
\mathbf{A}^T	transposed of \mathbf{A}
\mathbf{A}^{-1}	inverse of \mathbf{A}
a_i	element of \mathbf{a}
A_{ij}	element of \mathbf{A}
\dot{a}	first time derivative of a
\ddot{a}	second time derivative of a
a_t	partial time derivative of a
$\frac{da}{db}$	total derivative of a with respect to b
a_b	partial derivative of a with respect to b
\tilde{a}	approximation of a
$a^{(q)}$	a at q -th cycle

Matrix Calculation

Let \mathbf{b} be a column of n real variables, and $a(\mathbf{b}, t)$ be a scalar differentiable function of \mathbf{b} and t . Furthermore, let $\mathbf{a}(\mathbf{b}, t)$ be an m column of differentiable functions of \mathbf{b} and t , and $\mathbf{A}(\mathbf{b}, t)$ be an $m \times p$ matrix of differentiable functions of \mathbf{b} and t . Using i as row index and j as column index, the following matrix calculations are defined:

$$\begin{aligned} \dot{\mathbf{a}} &\equiv \left[\frac{da_i}{dt} \right]_{m \times 1} & \mathbf{a}_t &\equiv \left[\frac{\partial a_i}{\partial t} \right]_{m \times 1} & \dot{\mathbf{A}} &\equiv \left[\frac{dA_{ij}}{dt} \right]_{m \times p} & \mathbf{A}_t &\equiv \left[\frac{\partial A_{ij}}{\partial t} \right]_{m \times p} \\ \frac{da}{d\mathbf{b}} &\equiv \left[\frac{da}{db_j} \right]_{1 \times n} & a_{\mathbf{b}} &\equiv \left[\frac{\partial a}{\partial b_j} \right]_{1 \times n} & \frac{d\mathbf{a}}d\mathbf{b} &\equiv \left[\frac{da_i}{db_j} \right]_{m \times n} & \mathbf{a}_{\mathbf{b}} &\equiv \left[\frac{\partial a_i}{\partial b_j} \right]_{m \times n} \end{aligned}$$

List of Symbols

a	acceleration
a_c^m	maximum chassis acceleration
A	cross sectional area
b	design variable
\mathbf{b}	column of design variables
\mathbf{b}^I	column of intermediate design variables
\mathbf{b}^l	column of design variable lower bounds
\mathbf{b}^u	column of design variable upper bounds
\mathbf{b}_c	center point of the search subregion
\mathbf{b}_0	column of cycle start design
\mathbf{b}_*	column of cycle optimum design
\mathbf{b}_h^I	column of intermediate design variables of response function r_h
\mathbf{b}_{kh}^I	column of plan point intermediate design variables of response function r_h in the k -th design variable direction
\mathbf{b}_{0h}^I	column of intermediate design variables of response function r_h at start design \mathbf{b}_0
Δb	design variable change
c	constraint bound
c	damping coefficient
c_s	suspension damping
\mathbf{c}	column of constraint bounds
d	cross sectional diameter
d_{so}^m	maximum outward suspension deflection
d_t^m	maximum tire deflection
\mathbf{d}	column of directions of the search subregion
e_c	condition error
e_t	truncation error
\mathbf{e}	column of constraint approximation errors
E	maximum approximation error
E_f	objective function approximation error
E_g	maximum constraint approximation error
f	function
F	objective function
F_D	damping force
g	constraint
g^e	equivalent constraint
\mathbf{g}	column of constraints

h	index
HIC	Head Injury Criterion
i	index
j	index
k	index
k	stiffness coefficient
k_s	suspension stiffness
k_t	tire stiffness
l	length
m	mass
m	number of constraints
m_a	unsprung mass
m_c	sprung mass
\mathbf{m}	column of move limit factors
M	bending moment
\mathbf{M}	mass matrix
n	number of design variables
n_{mt}	number of (local) response maxima
n_t	number of time points
p	function
p	index
q	cycle number
q	iteration number
\mathbf{q}	column of generalized coordinates
\mathbf{q}_0	initial generalized positions
$\dot{\mathbf{q}}_0$	initial generalized velocities
$\Delta \mathbf{q}$	change of generalized variables
\mathbf{Q}^A	column of generalized applied forces
r	intermediate response variable, response function
\bar{r}	integral response function
\mathbf{r}	column of responses
\mathbf{s}	column of search subregion sizes
\mathbf{s}^l	column of search subregion lower bounds
\mathbf{s}^u	column of search subregion upper bounds
\mathbf{S}	column of search direction
t	time
t_0	initial time
t_f	final time
t_{mp}	time point for which a (local) response maximum occurs

t_p	time point
u	function
\bar{u}	integral function
v	function
v_D	relative damper velocity
\mathbf{v}	column of constraint violations
V	maximum constraint violation
x	position
x_G	desired slider-crank coupler point position
x_P	slider-crank coupler point position
y	position
y_a	wheel axle displacement
y_c	chassis displacement
y_G	desired slider-crank coupler point position
y_P	slider-crank coupler point position
\mathbf{y}	column of generalized positions
\mathbf{y}_0	column of initial generalized positions
\mathbf{z}	column of generalized velocities
\mathbf{z}_0	column of initial generalized velocities
α	Baumgarte's stabilization parameter
α	extrapolation factor
α^*	step in search direction
β	Baumgarte's stabilization parameter
β	penalty factor
β	relative finite difference step
β^c	column of damper curve parameters for compression
β^r	column of damper curve parameters for rebound
$\Delta\gamma$	crank rotation
γ	right-hand side of the acceleration equation
δ	Dirac delta function
ϵ_f	relative accuracy of function f
ζ	adjoint variable
ζ	column of adjoint variables
η	column of adjoint variables
θ	function
θ	rotation
κ^f	column of adjoint variables
λ	column of Lagrange multipliers
μ	adjoint variable

μ	column of adjoint variables
μ^f	column of adjoint variables μ at final time t_f
ν	right-hand side of the velocity equation
ξ	adjoint variable
ξ	column of adjoint variables
σ	stress
σ_a	maximum allowed stress
τ^f	column of adjoint variables
Φ	column of algebraic equations due to the kinematic couplings in a multibody system
Ψ	function
$\mathbf{0}$	column of zeros, matrix of zeros

Move Limit Parameters

errlrg	large approximation error
errmax	maximum approximation error
errsml	small approximation error
objacc	objective function accuracy
vioacc	constraint violation accuracy
violrg	large constraint violation
viomax	maximum constraint violation
viosml	small constraint violation

Chapter 1

Introduction

1.1 Analysis and Optimization

Nowadays, numerical analysis tools are commonly used in mechanical design. These tools allow mechanical engineers to design complex structures and machines. The first computer oriented analysis tools were developed in the sixties for structures, based on linear finite element methods (Bathe, 1996). Soon, finite element methods were applied in broader perspective like heat transfer and fluid flow, and developments were started for the analysis of nonlinear systems. At the end of the seventies, software packages became available for multibody systems. Multibody systems are mechanical models that consist of a finite number of rigid or flexible bodies. The governing equations are characterized by a strong geometric nonlinearity caused by the large displacements of motion. Furthermore, physical nonlinearities arising from, for example, contact or friction may be present. By now, a great variety of software is available that can automatically generate and solve the equations of motion for a wide range of multibody systems (Schiehlen, 1990).

Starting from an accurate numerical simulation, design optimization is a natural next step. Usually, a lot of design parameters do not have a predefined value, and in some cases even the topology is unknown. So, a convenient set of parameter values has to be selected that accomplish the design goals. Often, this is done manually by analyzing some different sets of design parameter values, and selecting the best set found. However, it is of great help for the design engineer to include design optimization tools in the analysis software for systematic design modification and improvement. This means that the design problem can be formulated as an optimization problem that can be solved by the computer.

Generally, an optimization problem is mathematically formulated as to find the

set of design variable values that will minimize an objective function subject to constraints. During the last decades several algorithms have been developed to solve such an optimization problem (Vanderplaats, 1984). The basic solution procedure consists of an iterative evaluation of objective function and constraints. However, with a numerical analysis involved, usually only few of them can be analytically expressed as function of the design variables. Most values of the objective function and constraints follow from the numerical analysis. Therefore, to extend the analysis software with design optimization facilities, an optimization algorithm has to be implemented and coupled with the analysis routines.

1.2 Structural Optimization

Numerical design optimization tools have been mainly developed in the field of finite element structural analysis. This is probably due to the early general acceptance of finite element methods for linear static and dynamic analysis of structures, and the successful development of the accompanying design sensitivity analysis (Haug *et al.*, 1986). Many optimization strategies applicable to structural optimum design require design sensitivity information, and therefore a numerically efficient and accurate sensitivity analysis is essential. Within this context, design variables are sizing parameters, such as cross sections of bars and beams, or shape parameters describing the global geometry of the design. A large amount of literature is available on structural sensitivity analysis. Reviews can be found in Adelman and Haftka (1986) and Haftka and Adelman (1989).

Several design optimization tools in structural design use a suitable approximation concept as interface between analysis routines and optimizer. The basic idea is to generate approximations of the objective function and constraints in a certain part of the design space, and to solve the optimum point for this approximate optimization problem. The approximate problem can be easily solved using a mathematical programming algorithm, since the approximate objective function and constraints can be cheaply evaluated. Various approximation concepts have been proposed (Barthelemy and Haftka, 1993). Many are based on analysis and design sensitivity data in a single point of the design space. Others use analyses in multiple design points to generate the approximations. The range of validity and the quality of the approximations determine whether several cycles of approximation and optimization are necessary to obtain the final optimum design.

Separation of numerical analysis and optimization by means of an approximation concept is computationally more convenient than a direct coupling, and avoids programming difficulties (Haftka and Gürdal, 1992). This is due to the fact that

mathematical programming algorithms often need many function value evaluations (i.e. numerical analyses), and require specific subroutines to evaluate the objective function and constraints. A second advantage is that the optimization is no longer a black-box. The design engineer can influence the optimization process, for example by selecting appropriate approximation models, or by reformulating the optimization problem after some cycles of approximation and optimization.

1.3 Optimization of Multibody Systems

Multibody analysis packages are generally capable of generating and solving algebraic relations and differential equations of motion for user-defined mechanical systems. Most packages, however, do not include routines that can automatically adjust the multibody systems parameters towards an improved design. To provide optimization facilities, the analysis code has to be extended with a suitable optimization strategy. The question is then how numerical analysis and optimization can be effectively combined into a general multibody design tool. Within this context, important aspects are the mathematical formulation of the optimization problem, the type of optimization algorithm to solve this problem, and the actual implementation. Altogether, they should guarantee a reliable and efficient design optimization for a wide range of multibody systems.

The multibody optimum design problem is defined by design variables, objective function and constraints. The design variables arise from the bodies, joints and force elements present in the multibody system. A wide variety of design variables can be distinguished such as the lengths of links, the sliding angle of a translational joint, or the stiffness and damping coefficients in a spring-damper element. The objective function and constraints are usually determined by the transient responses following from the numerical multibody analysis. Common multibody responses are displacements, velocities and accelerations, as well as induced forces and moments. For example, the objective may be to realize some specific motion subject to constraints on the maximum induced forces.

In literature concerning optimization of multibody systems, usually a sensitivity based optimization strategy is applied to a multibody system of fixed topology. Sensitivity based optimization algorithms have proven to be very effective for smooth problems with large numbers of design variables and constraints. Several successful applications have been reported for planar linkages design (Gabriele, 1993). Many have an ad hoc character and are concentrated on a specific type of mechanism. The multibody systems approach, however, requires the optimization to work in a more general framework, for both kinematics and dynamics.

Several authors have paid attention to the development of accurate and efficient design sensitivity methods for multibody systems, which they consider as the missing link between multibody analysis and optimization. For kinematically driven systems, Sohoni and Haug (1982) were one of the first to computer generate the governing equations for both analysis and sensitivity analysis. Dynamics was included by Haug *et al.* (1984). They studied the design sensitivity analysis of large-scale constrained dynamic systems. Ashrafioun and Mani (1990) proposed to symbolically instead of numerically generate the equations for both analysis and sensitivity analysis. Starting from symbolic formalisms, Bestle (1994) gave an extensive description of analysis, sensitivity analysis and optimization of multibody systems. For all references the optimization is finally (or is supposed to be) carried out by means of a direct coupling of the analysis and design sensitivity analysis routines with the selected mathematical programming algorithm.

1.4 Objective and Outline

This thesis studies the application of approximation concepts for the design optimization of multibody systems, instead of the usually applied direct coupling. The optimization can then benefit from the same advantages that have been mentioned for structural optimization in Section 1.2. Like finite element structural analysis, the numerical analysis of multibody systems is computationally expensive, especially for large complex systems. Approximation concepts may help to limit the required number of numerical analyses during the optimization. The open architecture of the approximate optimization process gives the design engineer access to the optimization, though still shielding the user from coming completely involved into the optimization loop. This opportunity to combine engineering experience with computer graphics and optimization may appear to be essential for an effective design optimization of multibody systems.

Starting point is the design optimization problem formulation. An appropriate formulation is one of the main preconditions for a successful optimization. Chapter 2 presents commonly used optimization problem formulations for multibody systems. It is not tried to be all inclusive in coverage. Instead, two analytical examples will illustrate some different formulations. The design parameterization is discussed by presenting some kinds of design variables. Next, several types of objective function and constraints are discussed. One of the important topics is how to deal with time dependent constraints. Optimization algorithms usually cannot deal directly with time dependent constraints, which means that time has to be removed, either by time discretization or integral type of functions.

Chapter 3 is the central part of this thesis, and describes the approximate optimization process. Shortly, different classes of approximation concepts are reviewed and their usefulness for multibody systems discussed. Then, the optimization problem is formulated such that approximation concepts can be incorporated. Design optimization software has been developed that is able to build and solve this approximate optimization problem. The two analytical multibody design examples of the previous chapter are optimized to illustrate the approach.

An accurate and cost-effective design sensitivity analysis is important if a sensitivity based approximation concept is applied. Such a concept has been used for the multibody design examples in Chapters 5 and 7, which were modeled in MECANO (Samtech, 1994) and DADS (CADSi, 1995), respectively. Finite differencing has been used to obtain the sensitivities with respect to the design variables, since both MECANO and DADS are not able to provide the required sensitivity information. However, finite difference sensitivity analysis suffers from two serious drawbacks: it is computationally expensive and accuracy problems may occur (Bestle and Eberhard, 1992). As mentioned in the previous section, alternative approaches for computing the gradient with additional equations have been published. Chapter 4 reviews the basic principles behind these methods, and discusses the consequences for an accompanying design sensitivity analysis of the proposed approximate optimization strategy.

Chapters 5, 6 and 7 show three multibody optimum design examples that have been solved by means of the developed design optimization software. Chapter 5 presents the weight minimization of a stress constrained four-bar mechanism. Flexibilities have been included in the MECANO model. In Chapter 6, a combined airbag and belt restraint system modeled with MADYMO (TNO, 1994) is designed for optimum crash safety. This problem clearly shows non-smooth behavior of objective function and constraints, and requires a special way to deal with the optimization problem since design sensitivities cannot be used. Chapter 7 deals with the design of a nonlinear damper of the suspension of a truck using DADS. For all examples approximate optimization proved to be an effective approach. Finally, Chapter 8 summarizes the main conclusions and gives some recommendations for further research.

1.5 A Guideline for the Reader

Two main parts can be distinguished: the theoretical basis in Chapters 2, 3 and 4, and the design applications in Chapters 5, 6 and 7. There is no need to read all chapters successively. After Chapter 3 one may proceed with any of the remaining

chapters. The chapter on design sensitivity analysis as well as each chapter on design application can be read separately. The only cross reference is made to Section 4.2 when finite difference sensitivities are used in Chapters 5 and 7.

Chapter 2

Optimization Problem Formulation

2.1 Introduction

Many mechanical systems can be modeled as a multibody system that is composed of a finite number of rigid or flexible bodies interconnected by joints and force elements. Well-known examples are road vehicles, mechanisms and robots. The bodies usually have mass and inertia, whereas the joints restrict the relative motion of interconnected bodies. The force elements introduce mechanical forces induced by, for example, springs and dampers. Due to mass, inertia and applied forces, second-order differential equations of motion arise. The joints add kinematic constraints, which are algebraic or first-order differential equations. The total set of equations is a mathematical representation of the multibody system, and can be numerically solved for simulation purposes.

Multibody system analysis software is able to automatically formulate and solve the governing equations of kinematics and dynamics for a wide range of multibody systems. Furthermore, graphical tools are provided to visualize the numerical analysis results. Several multibody packages are commercially available. Usually, three different types of analysis can be performed: kinematic, inverse dynamic and dynamic analysis (Haug, 1989). A kinematic analysis calculates the positions, velocities and accelerations of the generalized coordinates that follow from the prescribed driving conditions for the degrees of freedom. Afterwards, the forces to produce this motion can be obtained by an inverse dynamic analysis. Finally, dynamic analysis is concerned with the motion of the system that is due to the action of applied forces.

Computer simulation tools, like multibody analysis packages, enable design improvements early in the design stage. The engineer can, for example, analyze some different designs and select the best set of design variable values found. However, the numerical analysis of multibody systems is often computationally expensive. Combined with an increasing number of design variables, a 'many analyses' or 'trial and error' approach will become cumbersome and unsatisfactory. Here, an optimization tool can aid to find an improved design and reduce the required design time.

In engineering design, optimization problems are often written in the following mathematical format: find the set of n design variable values $\mathbf{b} \in \mathfrak{R}^n$ that will minimize the objective function:

$$F(\mathbf{b}) \tag{2.1}$$

subject to the inequality and equality constraints:

$$g_j(\mathbf{b}) \leq c_j, \quad j = 1, \dots, m', \tag{2.2}$$

$$g_k(\mathbf{b}) = c_k, \quad k = m' + 1, \dots, m, \tag{2.3}$$

within the design space:

$$b_i^l \leq b_i \leq b_i^u, \quad i = 1, \dots, n. \tag{2.4}$$

The scalar b_i is the i -th element of the design vector \mathbf{b} . The side-constraints define the design space, i.e. the region in which is searched for an optimum. For many multibody design examples there will be no equality constraints in the optimization problem. The kinematic constraints of the joints, and the equations of motion are *not* considered as equality constraints. This set of equations can be separately solved for the unknown state or response variables by means of the multibody analysis routines. Afterwards, the objective and constraint functions related to these response can be evaluated. Generally, only few functions of (2.1), (2.2) and (2.3) are explicitly known.

Various optimization algorithms are available to solve abovementioned standard optimization problem (Vanderplaats, 1984). Most of them are based on the iterative formula:

$$\mathbf{b}^q = \mathbf{b}^{q-1} + \alpha^* \mathbf{S}^q. \tag{2.5}$$

Starting from an initial design \mathbf{b}^0 , the old design \mathbf{b}^{q-1} is moved with step α^* in direction \mathbf{S}^q during the q -th iteration. Every algorithm has its own specific strategy to determine search direction \mathbf{S}^q and step size α^* . Well-known examples are

the gradient projection algorithm and sequential quadratic programming. The theoretical background can be found in many textbooks on optimization, either from an engineering or a mathematical point of view (e.g., Vanderplaats, 1984; Haftka and Gürdal, 1992; Gill *et al.*, 1981). The actual computer implementations can be obtained from several software libraries.

Transient responses play a central role in the optimization of multibody systems. A kinematic, dynamic or inverse dynamic analysis will usually give the behavior of the multibody responses as a function of time. This means that the objective function and constraints may be time dependent, which complicates the standard problem formulation of Equations (2.1) to (2.4). In the following two sections, the time dependent optimization problem is defined, and several different design variables, objective functions and constraints are presented. A separate section focuses on the time dependent constraints. Next, two small examples are given for which the response values can be analytically solved. They are used to illustrate the optimization problem formulation and will return in the subsequent chapters. Finally, this chapter is concluded with a discussion.

2.2 Optimization Problem

Within the time interval $t \in [t_0, t_f]$, the optimization problem is formulated as follows: find the set of design variable values $\mathbf{b} \in \mathcal{R}^n$ that will minimize the objective function:

$$F(\mathbf{b}) = F(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t), \quad t \in [t_0, t_f] \quad (2.6)$$

subject to the inequality constraints:

$$g_j(\mathbf{b}, t) = g_j(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t) \leq c_j \quad \forall t \in [t_0, t_f], \quad (2.7)$$

$$j = 1, \dots, m,$$

within the design space:

$$b_i^l \leq b_i \leq b_i^u, \quad i = 1, \dots, n. \quad (2.8)$$

The functions stored in the column $\mathbf{r}(\mathbf{b}, t)$ represent the transient responses calculated from the multibody governing equations. These are usually free or forced responses such as displacements, velocities, accelerations and joint forces. In the case of flexible bodies bending stresses may be considered. A classification of objective function and inequality constraints is given in Section 2.3. The equality

constraints have been omitted. If they do occur, they can be included by reformulation into inequality constraints (see e.g. Vanderplaats, 1984).

The design engineer has to formulate the multibody design problem as a mathematical optimization problem. This is an important stage in design optimization, for it has a direct influence on the optimization process and the final optimum design. A great variety of design variables, objective functions and constraints can be defined, depending on the specific multibody design application. In vehicle systems other design variables will be present than for linkages. Design for optimum crash worthiness gives rise to different objective functions and constraints compared with design for optimal comfort. The abovementioned formulation covers many different types of optimization problems that may occur for multibody systems.

The design parameterization plays a central role in the optimization problem formulation. Properly scaled design variables have to be selected that are not inter-related: a change in one design variable should not lead to a change of one of the other design variable values. For a fixed multibody system topology one can identify sizing and shape design variables. These design variables are usually related to the bodies, joints and force elements in the multibody system. Most of the multibody optimization literature focuses on sizing variables like stiffness and damping coefficients. Eberhard *et al.* (1996) applied shape optimization to nonlinear vehicle damper characteristics using Hermite splines to parameterize the damping curve.

Next, suitable objective function and constraints have to be selected that reflect the design objectives and specifications. This seems easy, but often there is more to it than meets the eye. The first difficulty that arises is which mathematical functions to use. Think for example of how to define comfort. Several different formulations may be used, such as peak, mean and cumulative measures on the acceleration response. After the mathematical functions have been defined, one has to distinguish the objective function from the constraints. Here, one can run into problems if a single objective function has to be selected from a couple of conflicting criteria. Multi-criteria optimization problem formulations may help (see e.g. Bestle, 1994), but they do not relieve the designer from the obligation to make decisions with respect to the multiple objective functions. Multi-criteria optimization is in fact an intermediate step offering structured aid to make a balanced choice towards an optimization problem with one objective function.

2.3 Objective Function and Constraints

Three classes of objective functions can be distinguished within the context of Equation (2.6): response independent, time point and integral type of functions.

A combination of the three is possible as well. A response independent objective function is determined by the design variables \mathbf{b} only:

$$F(\mathbf{b}) = f(\mathbf{b}). \quad (2.9)$$

The total mass of a linkage, for example, can usually be explicitly related to the link lengths and cross sectional areas.

The second type of objective function does depend on the multibody responses \mathbf{r} , but just at one or more specific time points t_p :

$$F(\mathbf{b}) = f(\mathbf{r}(\mathbf{b}, t_p), \mathbf{b}, t_p), \quad p = 1, \dots, n_t. \quad (2.10)$$

In Section 2.5.2 this function is illustrated by means of a slider-crank mechanism that has to generate a pre-defined path of motion. For some special design problems the time points t_p may also be design variables, or follow from certain preconditions:

$$\Psi_p(\mathbf{r}(\mathbf{b}, t_p), \mathbf{b}, t_p) = 0, \quad p = 1, \dots, n_t. \quad (2.11)$$

Such a condition is necessary if one wants to minimize the time to reach some kind of final state.

Finally, the third class of objective functions consists of integral type of functions:

$$F(\mathbf{b}) = \int_{t_0}^{t_f} f(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t) dt. \quad (2.12)$$

In control and vehicle design applications integral functions are commonly used. Initial time t_0 and final time t_f may depend on the design variables or be determined by a condition such as (2.11).

For the constraints, the same types of functions can be defined as well:

$$g(\mathbf{b}) = f(\mathbf{b}), \quad (2.13)$$

$$g(\mathbf{b}) = f(\mathbf{r}(\mathbf{b}, t_p), \mathbf{b}, t_p), \quad p = 1, \dots, n_t, \quad (2.14)$$

$$g(\mathbf{b}) = \int_{t_0}^{t_f} f(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t) dt. \quad (2.15)$$

Additionally, a time dependent constraint can be distinguished:

$$g(\mathbf{b}, t) = f(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t). \quad (2.16)$$

This type of time-varying constraint is parametric of nature, and often occurs in the optimization of multibody systems. Examples can easily found, like an acceleration that may never exceed some maximum value, or a spring whose deflection is limited. In some references time dependent constraints are called point-wise constraints (see e.g. Haug and Arora, 1979). The term point-wise will not be used in this thesis to avoid confusion with time point constraints.

A time dependent constraint also occurs if a max-value operator is removed from the objective function:

$$F(\mathbf{b}) = \max_{t \in [t_0, t_f]} \theta(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t). \quad (2.17)$$

The max-value operator may introduce discontinuities in the derivatives of the objective function. They can, however, be easily avoided by means of an artificial design variable. The original objective function (2.17) is then replaced by:

$$\bar{F}(\mathbf{b}) = b_{n+1}, \quad (2.18)$$

with an additional time dependent constraint:

$$g(\mathbf{b}, t) = \theta(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t) \leq b_{n+1} \quad \forall t \in [t_0, t_f]. \quad (2.19)$$

In textbooks on optimization, often this bound formulation is suggested to deal with min-max problems (see e.g. Haftka and Gürdal, 1992).

2.4 Dealing with Time Dependent Constraints

Mathematical programming algorithms generally cannot deal directly with parametric constraints like:

$$g(\mathbf{b}, t) \leq c \quad \forall t \in [t_0, t_f]. \quad (2.20)$$

Such a constraint has to be reformulated to remove the time dependence. The most straightforward way is to simply discretize the time interval into n_t time points. Then, the original constraint (2.20) is replaced by n_t constraints:

$$g_p(\mathbf{b}) = g(\mathbf{b}, t) \Big|_{t=t_p} \leq c, \quad t_p \in [t_0, t_f], \quad (2.21)$$

$$p = 1, \dots, n_t.$$

The time point distribution has to be sufficiently dense to avoid large constraint violations between two adjacent time points. As a consequence, discretizing time

dependent constraints can greatly increase the number of constraints, and thereby the cost of the optimization (Haftka and Gürdal, 1992).

Several equivalent constraint formulations have been proposed to remove the time dependence without increasing the number of constraints. In some references time dependent constraint (2.20) is replaced by an integral constraint function similar to:

$$g(\mathbf{b}) = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \max\{g(\mathbf{b}, t), c\} dt \leq c. \quad (2.22)$$

Constraint (2.22) will be satisfied as long as $g(\mathbf{b}, t)$ is smaller than or equal to the constraint bound c in the entire interval $[t_0, t_f]$. Whenever a violation occurs in between t_0 and t_f , the integral constraint will be violated as well, which means that the final optimum solution is not affected by the reformulation. Hsieh and Arora (1984) showed however, that from an optimization theory point of view, constraints (2.20) and (2.22) are different. This can be understood by noticing that an equivalent integrated constraint:

$$g^e(\mathbf{b}) = \int_{t_0}^{t_f} f(g(\mathbf{b}, t)) dt \quad (2.23)$$

represents the behavior of the time dependent constraint $g(\mathbf{b}, t)$ on the complete time domain $[t_0, t_f]$ by a single constraint value $g^e(\mathbf{b})$. Information is lost and as a consequence, equivalent constraints tend to blur design trends (Haftka and Gürdal, 1992). Due to the max-value operator, constraint (2.22) has an additional difficulty of discontinuously behaving gradients. Numerical difficulties are therefore likely to occur.

Both Grandhi *et al.* (1986) and Hsieh and Arora (1984) preferred to replace the original constraint (2.20) by critical time point constraints:

$$g_p(\mathbf{b}) = g(\mathbf{b}, t) \Big|_{t=t_{mp}} \leq c, \quad t_{mp} \in [t_0, t_f], \quad (2.24)$$

$$p = 1, \dots, n_{mt}.$$

Instead of a complete time discretization (2.21), they monitored the local maxima and the boundary maxima at t_0 and t_f of the time dependent constraint functions (see Figure 2.1). Part of these maxima will return as constraints in the optimization problem. Hsieh and Arora (1984) only retained the violated critical time points in the active set, where Grandhi *et al.* (1986) used a cutoff level to mark the important maxima. The time points for which the maxima occur depend on the design variable values:

$$t_{mp} = t_{mp}(\mathbf{b}), \quad p = 1, \dots, n_{mt}(\mathbf{b}), \quad (2.25)$$

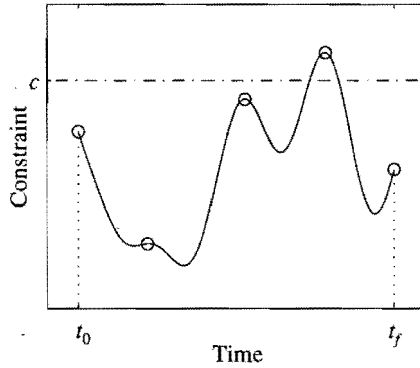


Figure 2.1: Candidate critical time point constraints.

and may shift during the optimization. This drift requires the critical time points to be frequently updated as the optimization progresses, for example after every iteration.

2.5 Examples

2.5.1 Impact-Absorber

An impact-absorber is modeled as a single degree of freedom system with a constant mass m of 1 kg, a linear spring k and a linear damper c (Figure 2.2). It is a simplified version of the nonlinear impact-absorber of Afimawala and Mayne (1974). At time $t = 0$ s, the initial mass position and velocity are $y(0) = 0$ m, and $\dot{y}(0) = 1$ ms⁻¹, respectively. Starting from the equation of motion and the initial conditions, the mass position as a function of time can be analytically solved:

$$y(k, c, t) = \begin{cases} \frac{e^{-t(c/2)}}{\sqrt{k-(c/2)^2}} \sin(t\sqrt{k-(c/2)^2}) & \text{if } 0 \leq \frac{(c/2)}{\sqrt{k}} < 1, \\ te^{-t\sqrt{k}} & \text{if } \frac{(c/2)}{\sqrt{k}} = 1, \\ \frac{e^{-t(c/2)}}{2\sqrt{(c/2)^2-k}} \left[e^{t\sqrt{(c/2)^2-k}} - e^{-t\sqrt{(c/2)^2-k}} \right] & \text{if } \frac{(c/2)}{\sqrt{k}} > 1. \end{cases} \quad (2.26)$$

The optimum design problem is to find the stiffness coefficient $b_1 = k$ Nm⁻¹ and the damping coefficient $b_2 = c$ Nsm⁻¹ that will minimize the maximum acceleration:

$$F(\mathbf{b}) = \max_{t \in [0, 12]} |\ddot{y}(b_1, b_2, t)| \quad (2.27)$$

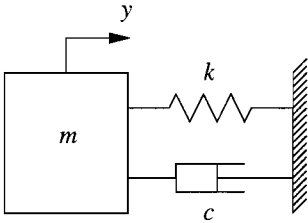


Figure 2.2: Impact-absorber.

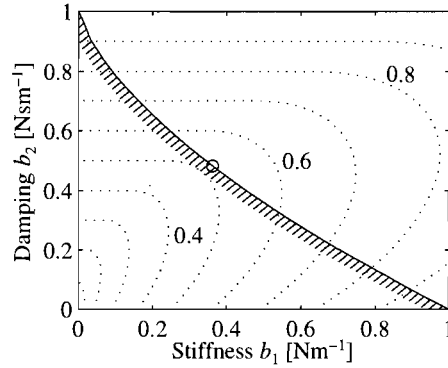


Figure 2.3: Optimization problem of the impact absorber for the maximum acceleration objective function. The optimum design is marked with \circ .

subject to the displacement constraint:

$$g(\mathbf{b}, t) = |y(b_1, b_2, t)| \leq 1 \quad \forall t \in [0, 12], \quad (2.28)$$

within the design space: $0 < b_1 < 1$ and $0 < b_2 < 1$. A time period of 12 s includes all important response maxima. In Figure 2.3 the optimization problem is visualized. The hatched line represents the displacement constraint bound, and the dotted lines represent contour lines of the maximum acceleration. The feasible region is in the upper right part of the design space. A single optimum solution is present, determined by the curvature of the maximum acceleration. For computational convenience the problem is reformulated into: minimize the artificial design variable:

$$F(\mathbf{b}) = b_3 \quad (2.29)$$

subject to the acceleration and displacement constraints:

$$g_1(\mathbf{b}, t) = \ddot{y}(b_1, b_2, t) \leq b_3 \quad \forall t \in [0, 12], \quad (2.30)$$

$$g_2(\mathbf{b}, t) = -\ddot{y}(b_1, b_2, t) \leq b_3 \quad \forall t \in [0, 12], \quad (2.31)$$

$$g_3(\mathbf{b}, t) = y(b_1, b_2, t) \leq 1 \quad \forall t \in [0, 12], \quad (2.32)$$

$$g_4(\mathbf{b}, t) = -y(b_1, b_2, t) \leq 1 \quad \forall t \in [0, 12], \quad (2.33)$$

with: $0 < b_1 < 1$, $0 < b_2 < 1$, and $b_3 \geq 0$.

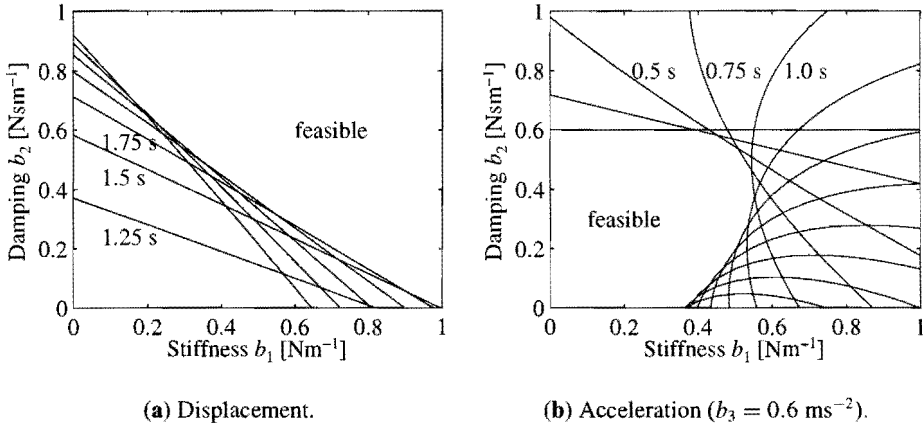


Figure 2.4: Time point constraints of the impact absorber.

b and $F(\mathbf{b})$	Initial design	Optimum design ^I	Optimum design ^{II}
b_1 [Nm ⁻¹]	1.0	0.3628	0.09850
b_2 [Nsm ⁻¹]	0.3	0.4823	0.7883
F^I [ms ⁻²]	0.8427	0.5203	
F^{II} [m ² s ⁻⁴]	0.1480		0.03804

I) maximum acceleration objective function

II) integral type of objective function

Table 2.1: Initial and optimum impact-absorber design.

Suppose the time dependence is removed by discretizing the time interval of 0 to 12 s. Figure 2.4 shows the displacement time point constraints, together with the acceleration time point constraints for $b_3 = 0.6 \text{ ms}^{-2}$, using a rather coarse discretization at every 0.25 s. For decreasing value of the artificial design variable the acceleration constraints shift towards the lower left corner, diminishing the feasible area until one point is obtained: the final optimum design. At the optimum, either two adjacent time point acceleration constraints and one time point displacement constraint will be active, or one acceleration and two displacement constraints. For 101 time points, the former case was found using the E04UCF nonlinear programming routine of the NAG-library (NAG, 1991). Initial and optimum design are tabulated in Table 2.1. To reach the final solution, 4 iterations, 5 function evaluations, and 5 gradient calculations were necessary (optimality and feasibility tolerance were set to 10^{-5}).

By comparing Figures 2.3 and 2.4, one can observe that the curvature of the

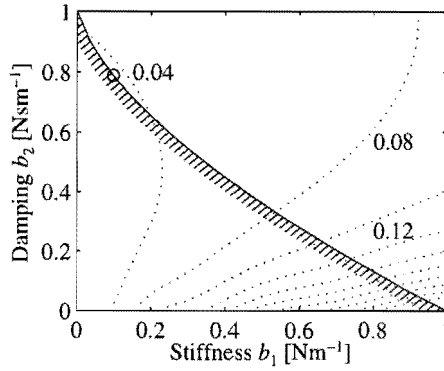


Figure 2.5: Optimization problem of the impact-absorber for the integral type of objective function. The optimum design is marked with \circ .

bounds of the feasible region is caused by the drift of the maxima of the displacement and acceleration functions. If the maxima had stayed at the same time, the feasible region would not have been determined by intersecting time point constraints. This confirms the necessity of a frequent updating if critical time points corresponding with the local maxima of the responses are used.

A second objective function is considered to illustrate the effect of a change in mathematical function on the final optimum solution. The maximum acceleration (2.27) is replaced by an integral type of objective function:

$$F(\mathbf{b}) = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \ddot{y}^2(b_1, b_2, t) dt. \quad (2.34)$$

In vehicle design applications, comfort is often defined by functions like (2.34) instead of the maximum acceleration (see e.g. Wimmer and Rauh, 1996). Figure 2.5 shows that the new objective function has a completely different behavior. The optimum design moves towards the upper left corner of the design space (see Figure 2.5). Now, 14 iterations with 23 function and 23 gradient calculations yielded the final optimum design.

2.5.2 Slider-Crank Mechanism

Paradis and Willmert (1983) described a mechanism design problem with explicitly known objective function and constraints. A four-bar slider-crank mechanism has to be designed such that a desired coupler curve is generated. Figure 2.6 shows the four-bar mechanism with eight design variables b_i , and the coupler point P that

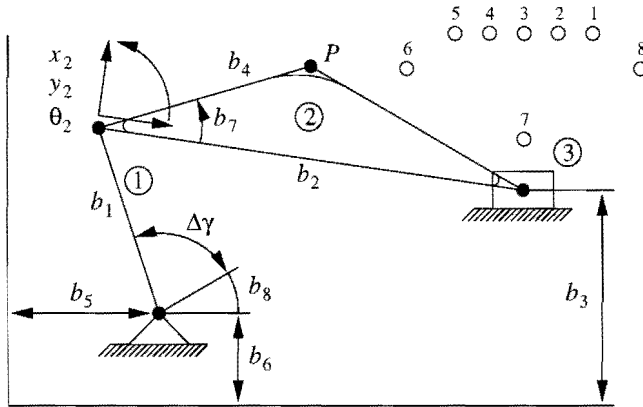


Figure 2.6: Slider-crank mechanism and coupler curve.

Point k	1	2	3	4	5	6	7	8
$x_G(k)$	26	23	20	17	14	10	20	30
$y_G(k)$	16	16	16	16	16	13	7	13
$\Delta\gamma(k)$ in degrees	0	22	44	66	88	129	221	314

Table 2.2: Path coordinates and prescribed timing.

should generate the desired curve. The prescribed path and timing of the coupler curve is given in Table 2.2. The desired (x_G, y_G) positions of P are tabulated as a function of the crank rotations $\Delta\gamma(k)$ relative to the starting angle b_8 .

The optimization problem is to minimize the objective function:

$$F(\mathbf{b}) = \sum_{k=1}^8 \{x_P(\Delta\gamma(k)) - x_G(k)\}^2 + \{y_P(\Delta\gamma(k)) - y_G(k)\}^2 \quad (2.35)$$

subject to the movability constraints:

$$g_1(\mathbf{b}) = -0.85b_2 + b_1 + b_3 - b_6 \leq 0, \quad (2.36)$$

$$g_2(\mathbf{b}) = -0.85b_2 + b_1 - b_3 + b_6 \leq 0, \quad (2.37)$$

and design space bounds: $1 \leq b_1 \leq 30$, $1 \leq b_2 \leq 30$ and $1 \leq b_4 \leq 30$. The movability constraints ensure that the linkage can operate for the complete range of crank rotations. The position of coupler point $P(x_P, y_P)$ follows from the response variables x_2, y_2 and θ_2 of body 2, and the design variables b_4 and b_7 :

$$x_P = x_2 + b_4 \cos(b_7 + \theta_2), \quad (2.38)$$

$$y_P = y_2 + b_4 \sin(b_7 + \theta_2). \quad (2.39)$$

Usually, the state or response variable values have to be numerically solved from the governing equations of the multibody system. But for the slider-crank mechanism x_2 , y_2 and θ_2 can be analytically derived as a function of rotation $\Delta\gamma$:

$$x_2 = b_1 \cos(b_8 + \Delta\gamma) + b_5, \quad (2.40)$$

$$y_2 = b_1 \sin(b_8 + \Delta\gamma) + b_6, \quad (2.41)$$

$$\theta_2 = \arcsin\left(\frac{b_3 - y_2}{b_2}\right). \quad (2.42)$$

Objective function (2.35) clearly is of the second class of functions defined by Equation (2.10).

b and $F(\mathbf{b})$	Initial design	Optimum design
b_1	6	9.388
b_2	25	30.00
b_3	-4	-6.650
b_4	11	12.71
b_5	3	8.224
b_6	4	9.462
b_7	2	0.8215
b_8	2	0.9160
F	3486	3.746
g_1	-23.25	-32.224
g_2	-7.25	0.0

Table 2.3: Initial and optimum slider-crank design.

In Table 2.3 the initial design and corresponding optimum solution found by the NAG E04UCF routine are given. The design variable values of b_7 and b_8 are in radians. The solution was obtained after 36 iterations, 44 function evaluations and 44 gradient calculations. The optimum design nearly exactly corresponds with the solution of Gabriele (1993).

2.6 Discussion

A well-considered definition of design variables, objective function and constraints is of vital importance for a successful design optimization. From a modeling stand of view, the optimization problem should be a good representation of the mechanical design problem. The impact-absorber example has illustrated that a simple

change of the objective function may lead to a completely different optimum solution. Furthermore, numerical difficulties inherent to the optimization problem formulation should be avoided. This numerical stand of view includes a proper design parameterization, scaling of the design variables and constraints, and selection of computationally convenient objective function and constraints. As an example, the difficulties arising from equivalent integrated constraints have been mentioned.

For the examples of Section 2.5, the multibody responses could be analytically solved. As a result, the user subroutines, as required by the NAG optimization algorithm, could be easily generated. However, for larger and more complex design applications this ad hoc approach is not an option anymore. Ideally, optimization facilities should be included in the multibody analysis tool. This means that a coupling between analysis and optimization has to be established, with a designer able to specify the precise relations. In the next chapter it is proposed to realize this coupling by means of approximation concepts.

Chapter 3

Sequential Approximate Optimization

3.1 Introduction

Multibody system governing equations have typical features that complicate the numerical optimization. First of all, the equations include time, and therefore a time dependent optimization problem follows, which cannot be directly solved by standard mathematical programming algorithms. Furthermore, the multibody equations are highly nonlinear. This often results in a computationally expensive solution, especially for larger engineering applications. Additionally, design sensitivity methods successfully developed for linear finite element problems cannot be applied to multibody systems just like that.

An optimization tool for multibody systems should work for most of the mechanical designs that can be modeled from the multibody elements library. Think for example of mechanical systems composed of linkages, nonlinear dampers, or cams. This requires a rather robust optimization strategy, integrated with the numerical analysis. Besides, the multibody software should be extended with graphical means for a user-friendly formulation of the optimization problem, and a clarifying visualization of the optimization results. This thesis concentrates on the former aspect: the optimization strategy.

The numerical and computationally expensive nature of many multibody responses places great demands on the optimization strategy. In any case, the number of numerical analyses required during the optimization should be limited. A call for an objective function or constraint evaluation at a certain design point means a new multibody analysis to be carried out. Furthermore, it is advantageous when

the user is able to control the optimization process. Then, one can keep an eye on the progress of the optimization, and intervene if necessary, for example by an adjustment of the multibody system model, or by a redefinition of the optimization problem formulation.

Approximation concepts give the opportunity to accomplish the demands for an effective design optimization of multibody systems. They can form an interface in between analysis and optimization. The basic idea is to build approximations of response based objective function and constraints that can be easily evaluated by the optimizer, without any call to the numerical analysis. Such an interface may open new ways to make the design optimization of multibody systems more accessible and controllable compared with a direct coupling of analysis and optimization.

This chapter introduces the approximate optimization strategy in the context of multibody systems. Several different approximation concepts are discussed, together with the accompanying intermediate design variables and responses. Next, the optimization problem is formulated such that approximation concepts can be incorporated. Starting from a sequence of sensitivity based approximations, a design optimization tool is presented that has been especially developed for multibody systems. If the design sensitivities cannot be calculated at a reasonable accuracy, a sequence of multi-point approximations is generated instead. The impact-absorber and slider-crank mechanism, presented in the previous chapter, are used to illustrate the effectiveness of the approach.

3.2 Approximation Concepts

Starting point of the approximate optimization strategy is the generation of approximations for objective function and constraints related to numerically expensive responses. Analysis and often also design sensitivity analysis results at one or more design points are used to build the approximation functions. The original optimization problem is then replaced by an approximate problem that is explicitly known or easily calculable, and that can be solved within the region the approximation is valid. Barthelemy and Haftka (1993) reviewed both the basic and more recently developed approximation concepts in structural optimization. They distinguished local, global and mid-range concepts.

Local approximations of objective function and constraints are based on function values and derivative values with respect to the design variables in a single point of the design space, for example linear or reciprocal approximations. Usually, these approximations are only valid in the vicinity of this design point. Therefore, a search subregion is defined in which the approximate optimization problem

is solved. A sequence of approximate optimization cycles has to be performed to reach the final optimum solution. Local approximation concepts are very popular, because large numbers of design variables and constraints can be handled without great difficulty. For multibody systems this is an important advantage since many constraints may occur, especially if time discretization is used to deal with time dependent constraints. Moreover, efficient and accurate design sensitivity analysis methods for multibody systems have shown major progress during the last decade (see Chapter 4). Both aspects indicate that local approximation concepts can be effectively used for the optimization of multibody systems.

Global objective function and constraint approximations are built from analysis runs spread over the design space. The original optimization problem is then completely substituted by an approximate problem formulation, which means that numerical analysis and optimization are decoupled. Within this framework response-surface techniques are commonly applied (Schoofs, 1987): approximate model functions are fitted to the results of a set of numerical experiments corresponding to a carefully selected experimental design. Usually, an iterative updating of the model functions and the experimental design follows. Nevertheless, it is often rather difficult to find the correct model functions. Sacks *et al.* (1989) tried to solve this problem by means of a new model building strategy for the deterministic computer experiments. Though, the author of the present thesis was not convinced that this strategy improves the response-surface strategy (Etman, 1994). Additionally to the model selection problem, global approximations become inefficient for growing number of design variables and constraints. These two drawbacks limit the application for multibody systems. Therefore, global approximations are not studied any further in this thesis. Still, they may be useful in the preliminary optimization stage, to gain more insight in the behavior of some important response functions or to search for promising design regions.

Finally, mid-range approximations are extended local approximations. As a consequence, the optimization is a sequential approximate optimization process. However, the approximations are based on response data at more than one design point. One can think of using analysis and design sensitivity analysis data of former design cycles to improve the current cycle (i.e. the so-called single-point-path mid-range approximations). Another strategy is to generate a plan of experiments of several design points within the search subregion of each design cycle (multi-point-path). Both the single-point-path as well as the multi-point-path mid-range concepts are valuable for the optimization of multibody systems. The single-point-path concepts benefit from the same advantages that have been mentioned above for the local concepts. The multi-point-path concepts may be effective if, due to inaccuracies, design sensitivities cannot be used to build the approximations.

3.3 Intermediate Variables and Responses

Introduction of intermediate design variables and intermediate response quantities aims at creating a high quality approximation that yields an efficient and reliable optimization process (Vanderplaats, 1993). The key idea is to improve an approximation of objective function or constraint by incorporating nonlinear behavior that is explicitly known or physically present. For truss-structures, a well-known example is to linearly approximate a stress constraint with respect to the reciprocal of the cross-sectional area. The stress as function of the reciprocal area behaves more linearly compared with the stress as function of the area itself. Thus the approximation can be improved if this intermediate design variable is accounted for. Another example is the relationship of the bending stress in a beam as function of the bending moment and the physical dimensions. The stress is highly nonlinear with respect to design variables such as the beam height, width or diameter (see e.g. Equation (5.1) in Chapter 5). Therefore, it is beneficial to approximate the bending moments instead of the stresses to obtain approximations for the stress constraints. Now the term intermediate response quantity is used.

Approximate optimization strategies are commonly applied to structures. However, for multibody systems intermediate design variables and intermediate responses are hardly mentioned. Though, the basic principle is not restricted to any specific area of application whatsoever. For multibody systems at least the same potential is present. Intermediate design variables and responses equivalent with the structural case are quite easily identified, like bending moments in beams. But also many relations exist that specifically belong to multibody systems. In design of machinery objective function and constraints are often operations on the numerical responses following from the multibody analysis. Such operations are usually explicitly known or can be easily evaluated, which directly points towards intermediate response quantities. The objective function of the slider-crank mechanism in Section 2.5.2 is, for example, a sum of squares on the position of point P . Calculation of a comfort criterion such as (2.34) is also rather straightforward, given a column of accelerations in time.

Selection of suitable intermediate design variables and intermediate response quantities heavily relies on the physical insight of the design engineer. The author's opinion is that this is far more beneficial than a drawback. Optimization will steadily become an integrated part of the analysis, and will probably never become just a press of a button. The optimization software should provide graphical tools to visualize the approximate optimization process and, more specifically, to enable the engineer to introduce intermediate design variables and responses. Then, the engineer has at his disposal a design tool that can combine both computer power

and engineering experience.

3.4 Approximate Optimization Problem

Sequential approximate optimization promises to be an effective approach for multi-body systems. One can use a local approximation concept for smooth problems, or switch to a multi-point mid-range concept whenever sensitivities cannot be accurately calculated. For both concepts, the optimization problem defined in Section 2.2 of the previous chapter is replaced by a sequence of explicitly known approximate optimization problems. The approximate optimization problem of the q -th cycle is defined as: minimize the approximate objective function:

$$\tilde{F}^{(q)}(\mathbf{b}) = F(\tilde{\mathbf{r}}^{(q)}(\mathbf{b}, t), \mathbf{b}, t), \quad t \in [t_0, t_f] \quad (3.1)$$

subject to the approximate constraints:

$$\begin{aligned} \tilde{g}_j^{(q)}(\mathbf{b}, t) = g_j(\tilde{\mathbf{r}}^{(q)}(\mathbf{b}, t), \mathbf{b}, t) \leq c_j \quad \forall t \in [t_0, t_f], \\ j = 1, \dots, m, \end{aligned} \quad (3.2)$$

within the search subregion:

$$s_i^{l(q)} \leq b_i \leq s_i^{u(q)}, \quad i = 1, \dots, n. \quad (3.3)$$

As in Section 2.3, the functional relations can be either response dependent, response independent, time dependent, or time independent. Time discretization is used to remove time dependence (see Section 2.4). Then the approximate optimization problem has been reduced to the standard form of Equations (2.1) to (2.4).

The framework of the approximate optimization problem is depicted below:

$$\begin{array}{ccc} \mathbf{b} & \xrightarrow{\text{explicit}} & \tilde{F}, \tilde{\mathbf{g}} \\ \text{explicit} \downarrow & & \uparrow \text{explicit} \\ \mathbf{b}^I & \xrightarrow{\text{approximation}} & \tilde{\mathbf{r}} \end{array} \quad (3.4)$$

Herein, two types of objective function and constraints are distinguished: objective function and constraints are either completely explicitly known, or are related to the approximate responses $\tilde{\mathbf{r}}$. In the latter case, the relationship between the objective function and constraints on the one hand, and the multibody responses on the other hand, is supposed to be explicitly known or easily calculable. So, multibody responses \mathbf{r} are treated as intermediate response quantities that are linearly

approximated with respect to the intermediate design variables \mathbf{b}^l . The relationship between intermediate design variables \mathbf{b}^l and design variables \mathbf{b} is supposed to be explicitly known.

The approximate optimization problem has to be solved by an appropriate mathematical programming algorithm. When intermediate design variables and intermediate response quantities are used, usually a smooth but nonlinear programming problem follows. Since only a limited part of the design space is considered, the approximate optimization problem will often have only one optimum solution. However, the approximate optimization problem is not guaranteed to be convex, and therefore more than one local optimum solution may occur. Sepulveda and Schmit (1993) proposed to use a global optimizer for the solution of the approximate optimization problem to yield a globally convergent optimization strategy. For specific types of intermediate design variables convexity can be proved, and in that case an optimizer can be selected that utilizes this quality of the approximation. Well-known examples are convex linearization (Fleury and Braibant, 1986) and the method of moving asymptotes (Svanberg, 1987, 1996).

Within the search subregion, the approximate optimization problem does not necessarily have a feasible optimum solution. Occasionally, the initial design may have one or more violated constraints. The search subregion defined around this infeasible starting point may still not contain a feasible solution, which means that it is completely situated inside the infeasible domain. The objective is then to first move towards a feasible solution before minimizing the objective function itself. Many nonlinear programming packages indeed start to minimize the maximum constraint violation instead of the original objective function. Haftka and Gürdal (1992) suggested to minimize:

$$\bar{F}^{(q)}(\mathbf{b}) + \beta b_{n+1} \quad (3.5)$$

subject to the constraints:

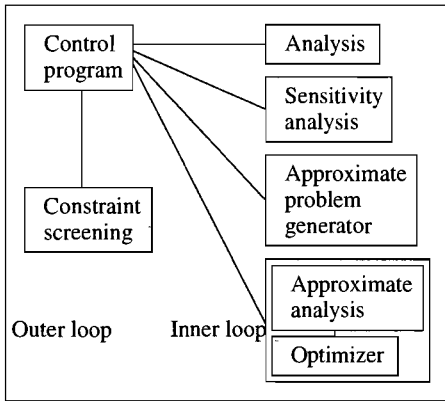
$$\begin{aligned} \tilde{g}_j^{(q)}(\mathbf{b}, t) - b_{n+1} \leq c_j \quad \forall t \in [t_0, t_f], \\ j = 1, \dots, m, \end{aligned} \quad (3.6)$$

within the search subregion:

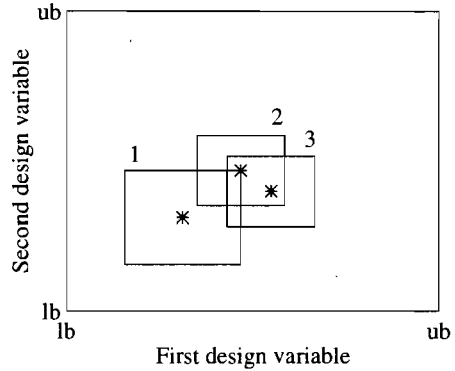
$$s_i^{l(q)} \leq b_i \leq s_i^{u(q)}, \quad i = 1, \dots, n, \quad (3.7)$$

$$b_{n+1} \geq 0, \quad (3.8)$$

instead of the original optimization problem (3.1) to (3.3). Design variable b_{n+1} takes care of the constraint relaxation. The value of β should be sufficiently large to guarantee that the reduction of b_{n+1} is emphasized over the minimization of $\bar{F}^{(q)}$. Then, constraint relaxation will only occur if there is no alternative.



(a) Program structure, Vanderplaats (1993)



(b) Sequence of search subregions. Lower and upper bounds are denoted by lb and ub, respectively.

Figure 3.1: Sequential approximate optimization using a local approximation concept.

3.5 Design Optimization Tool for Multibody Systems

3.5.1 Approach

In the current research a design optimization tool has been developed based on local approximations. It has been especially designed for multibody systems, and covers time dependent constraints. The optimization tool does not include multibody analysis routines and has to be coupled with an external software package, such as MECANO (Samtech, 1994) and DADS (CADSi, 1995). Most commercial multibody packages do not provide the derivatives of the multibody responses with respect to the design variables. In that case, the gradients necessary for the approximations are obtained by means of finite differencing.

On the basis of Figure 3.1, Vanderplaats (1993) described the main program structure of the sequential approximate optimization process for a local approximation concept. He concentrated on finite element structural analysis, but the program structure is the same for multibody analysis. The process starts with the analysis of the initially proposed design, followed by an evaluation of all constraint functions (constraint screening). Approximation models are generated for the critical and potentially critical constraints based on analysis and design sensitivity data. The approximate optimization problem is built, and the region of validity is bounded by so-called move limits. Within this search subregion, the approximate problem is iteratively solved by the optimizer. At the calculated optimum a new design cycle

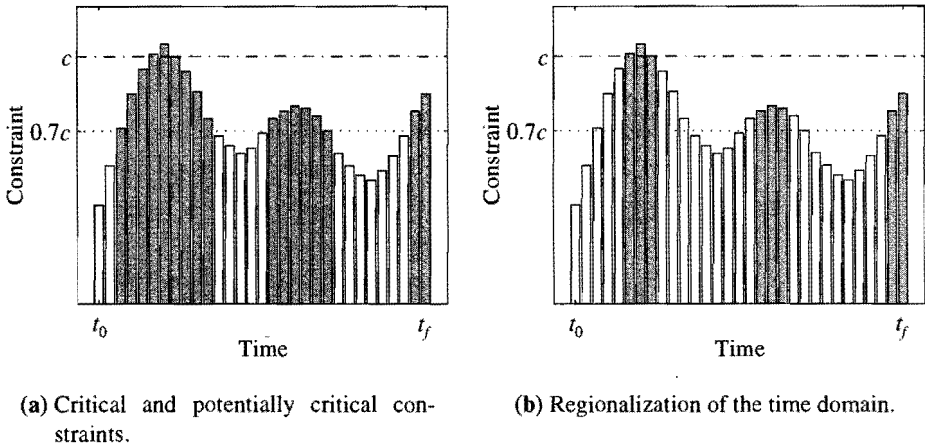


Figure 3.2: Constraint deletion.

can be started, as visualized in Figure 3.1(b). So, the term cycle is used for the outer loop, and the term iteration for the inner loop.

3.5.2 Constraint Deletion

The constraint screening tries to identify the important constraints in the optimization problem. Constraints that are not critical or potentially critical at the cycle start design $\mathbf{b}_0^{(q)}$ are removed. This can greatly reduce the number of constraints in the approximate optimization problem, and thereby the cost of the numerical optimization. Additionally, design sensitivity information is only required for the retained critical and potentially critical constraints. This gives the opportunity to reduce the cost of the design sensitivity analysis (see the next chapter).

Great potential exists for constraint deletion if time dependent constraints are replaced by time point constraints. All constraints can be removed whose value at $\mathbf{b}_0^{(q)}$ is smaller than e.g. 70% of the constraint bound (Figure 3.2(a)). Furthermore, for each local maximum of the constraint $g(\mathbf{b}, t)$, only a few time point constraints $g_p(\mathbf{b}, t_p)$ near the maximum have to be retained (Figure 3.2(b)). This is comparable with the regionalization of Vanderplaats (1993) who considered just a few stress constraints in specific regions of a structure, since many nearby elements are expected to have about the same stress value. The developed optimization tool allows the user to define whether a regionalization of the time domain is applied, and how many time points are added before and after the local maxima, including initial and final time. In the examples usually two points before and two after the maxima above 70% of the constraint bound are selected, unless otherwise stated.

Hsieh and Arora (1984) only retained the constraint maxima. Besides these worst time points no further points were considered. They studied an equation alike:

$$g(\mathbf{b}, t) \Big|_{t=t_{mp}} = g(\mathbf{r}(\mathbf{b}, t), \mathbf{b}, t) \Big|_{t=t_{mp}} \leq c. \quad (3.9)$$

The time point for which a maximum occurs depends on the design variables: $t_{mp} = t_{mp}(\mathbf{b})$. Taking the total derivative with respect to a design variable $b \in \mathbf{b}$ yields:

$$\begin{aligned} \frac{dg}{db} &= \frac{\partial g}{\partial r} \Big|_{t=t_{mp}} \left[\frac{\partial r}{\partial b} \Big|_{t=t_{mp}} + \frac{\partial r}{\partial t} \Big|_{t=t_{mp}} \frac{dt_{mp}}{db} \right] + \frac{\partial g}{\partial t} \Big|_{t=t_{mp}} \frac{dt_{mp}}{db} + \frac{\partial g}{\partial b} \\ &= \left[\frac{\partial g}{\partial r} \frac{\partial r}{\partial b} \right]_{t=t_{mp}} + \left[\frac{\partial g}{\partial r} \frac{\partial r}{\partial t} + \frac{\partial g}{\partial t} \right]_{t=t_{mp}} \frac{dt_{mp}}{db} + \frac{\partial g}{\partial b}. \end{aligned} \quad (3.10)$$

The second term vanishes at a local maximum since the total derivative of g with respect to t is zero:

$$\frac{dg}{dt} = \left[\frac{\partial g}{\partial r} \frac{dr}{dt} + \frac{\partial g}{\partial t} \right]_{t=t_{mp}} = \left[\frac{\partial g}{\partial r} \frac{\partial r}{\partial t} + \frac{\partial g}{\partial t} \right]_{t=t_{mp}} = 0. \quad (3.11)$$

So a change of the point t_{mp} has no effect on $\frac{dg}{db}$. Hsieh and Arora (1984) used this observation to treat the maximum time points as fixed during an iteration of their optimization algorithm. Although the shift of the top position has only a second order effect on the constraint value, it does contribute to the curvature (see for example Figure 2.4). If the optimum is determined by a curvature which the approximation does not predict, oscillations will occur during the approximate optimization process. Some additional time points near the local maximum can solve this problem: the curvature is represented by intersecting time point constraint approximations. A conservative approximation at the maximum might be an interesting alternative to avoid the increase of the number of time points in the approximate optimization problem.

3.5.3 Approximation

All responses related to the retained active and potentially active constraints need to be approximated. For each design cycle they are linearly approximated with respect to the intermediate design variables at the cycle start design. The linear approximation of a time independent response function $r_h(\mathbf{b})$ can be written as:

$$\tilde{r}_h(\mathbf{b}_h^I) = r_h(\mathbf{b}_{0h}^I) + \sum_{k=1}^n (b_{hk}^I - b_{0hk}^I) \left(\frac{\partial r_h}{\partial b_{hk}^I} \right)_{\mathbf{b}_{0h}^I}. \quad (3.12)$$

Herein, superscript (q) has been omitted. For a time dependent response function $r_h(\mathbf{b}, t)$ the approximation at a certain time point t_p is given by:

$$\tilde{r}_h(\mathbf{b}_h^I, t_p) = r_h(\mathbf{b}_{0h}^I, t_p) + \sum_{k=1}^n (b_{hk}^I - b_{0hk}^I) \left(\frac{\partial r_h}{\partial b_{hk}^I} \right)_{\mathbf{b}_{0h}^I, t_p}. \quad (3.13)$$

Each response function $r_h(\mathbf{b})$ or $r_h(\mathbf{b}, t)$ may have its own intermediate design variables $\mathbf{b}_h^I(\mathbf{b})$. The optimization tool allows the following type of intermediate design variables:

$$b_{hk}^I = b_{hk}^I(b_k), \quad k = 1, \dots, n. \quad (3.14)$$

Cross-relations in between design variables cannot be defined.

The number of responses to be approximated is highly influenced by the optimization problem formulation. Suppose a constraint is active that is an integral function of a response: $g(\mathbf{b}) = \int f(r(\mathbf{b}, t))dt$. If r is indeed treated as an intermediate response variable, then all discrete responses $r_p(\mathbf{b}, t_p)$ from t_0 to t_f have to be approximated, or a large part of them, to calculate the integral. This can be avoided by directly approximating constraint g without the introduction of an intermediate response variable. In that case, however, any nonlinearity implied in the integral formulation is not included anymore in the approximation, which may cause the optimization to behave less well. Here, the cost to calculate the derivatives of all necessary responses should be weighed against the efficiency and robustness of the optimization process. In any case, integral functions should only be included in the optimization problem if they arise from a design point of view. They should not be introduced just to remove the time dependency.

For each optimization cycle, the quality of the approximations can be checked by comparing the approximate objective and constraint values for the newly proposed design with the corresponding multibody analysis values. Differences between the approximated and calculated values are measures of the quality of the generated approximations. So, after the q -th cycle has been completed, the following approximation error can be calculated for the objective function:

$$E_f^{(q)} = \left| \frac{\tilde{F}^{(q)}(\mathbf{b}_*^{(q)}) - F(\mathbf{b}_*^{(q)})}{F(\mathbf{b}_*^{(q)})} \right| \times 100\%, \quad (3.15)$$

where $\mathbf{b}_*^{(q)}$ is the proposed optimum design computed from the approximate optimization problem of the q -th cycle. The maximum constraint approximation error is given by:

$$E_g^{(q)} = \max_{j=1, \dots, m} e_j^{(q)} \quad (3.16)$$

with:

$$e_j^{(q)} = \begin{cases} \left| \frac{\hat{g}_j^{(q)}(\mathbf{b}_*^{(q)}) - g_j(\mathbf{b}_*^{(q)})}{g_j(\mathbf{b}_*^{(q)})} \right| \times 100\% & \text{if } g_j \text{ is time independent,} \\ \max_{p=1, \dots, n_t} \left| \frac{\hat{g}_j^{(q)}(\mathbf{b}_*^{(q)}, t_p) - g_j(\mathbf{b}_*^{(q)}, t_p)}{g_j(\mathbf{b}_*^{(q)}, t_p)} \right| \times 100\% & \text{if } g_j \text{ is time dependent.} \end{cases} \quad (3.17)$$

Many of the (time point) constraints will not contribute to this error $E_g^{(q)}$ since they have been removed from the approximate optimization problem after the constraint screening. In the same way the maximum constraint violation can be calculated:

$$V^{(q)} = \max_{j=1, \dots, m} v_j^{(q)} \quad (3.18)$$

with:

$$v_j^{(q)} = \begin{cases} \frac{g_j(\mathbf{b}_*^{(q)}) - c_j}{c_j} \times 100\% & \text{if } g_j \text{ is time independent,} \\ \max_{p=1, \dots, n_t} \frac{g_j(\mathbf{b}_*^{(q)}, t_p) - c_j}{c_j} \times 100\% & \text{if } g_j \text{ is time dependent.} \end{cases} \quad (3.19)$$

Both the approximation errors and the constraint violation are used in the move limit strategy presented in the next section. For their calculation the objective function value as well as the constraint bounds should be non-zero.

3.5.4 Move Limit Strategy

A move limit strategy has to determine the size of the search subregion at the start of each new design cycle. This strategy has a great influence on the efficiency of the optimization process. A correct choice of the search subregion is important for a good convergence of the optimization process. Large move limits can cause the solution process to oscillate, while small move limits may slow down the convergence. The effect of the move limit strategy is directly related to the quality of the approximations. Poor approximations much more need the support of the move limit strategy during the optimization than high quality approximations. Unfortunately, move limit strategies are often poorly outlined in literature. Below, the main elements of the implemented move limit strategy are described at length.

Usually, the solution $\mathbf{b}_*^{(q)}$ of the approximate optimization problem will be the starting point of the next cycle $\mathbf{b}_0^{(q+1)}$. The proposed cycle start design is only rejected if the maximum approximation error has become too large or too high an infeasibility occurs, starting from a feasible or nearly feasible design. In that case

the current cycle is repeated for a smaller search subregion. The maximum allowed approximation error err_{max} and infeasibility $viomax$ have been set to 40% and 10%, respectively.

During the optimization, the sizes of the search subregion are adapted depending on the reported maximum approximation errors, maximum constraint violations, and the convergence behavior of the design variables. Search subregion upper and lower bounds can be calculated from:

$$s_i^{l(q)} = b_{0i}^{(q)} - 1/2s_i^{(q)}, \quad i = 1, \dots, n, \quad (3.20)$$

$$s_i^{u(q)} = b_{0i}^{(q)} + 1/2s_i^{(q)}, \quad i = 1, \dots, n. \quad (3.21)$$

Herein, the size of the search subregion is defined relatively to the magnitude of the design variables:

$$s_i^{(q)} = 2m_i^{(q)} \left| b_{0i}^{(q)} \right|, \quad i = 1, \dots, n, \quad (3.22)$$

which means that the design variable values should not become zero. As an alternative, Equation (3.27) may be used. The search subregion can be adjusted by means of move limit factors $m_i^{(q)}$. So, besides an initial design also initial move limit factors have to be selected. For the following cycles, the move limit strategy increases or decreases the move limit factors according to the optimization history. Move limits are decreased for approximation errors exceeding $err_{lrq} = 25\%$ or constraint violations rising above $viol_{lrq} = 7.5\%$. If the approximation is highly accurate ($<err_{sml}=10\%$) and the constraint violation remains small ($<viom_{sml}=5\%$), in that case the move limits are increased.

Near the optimum a higher accuracy is desired. This final optimization stage is often marked by oscillations or small changes in objective function value. If one of these conditions is met (see Appendix A for the computer implementation), then the move limit factors of oscillating design variables are halved. For an optimum constrained by a total amount of constraints that equals the number of design variables, usually very fast convergence occurs, without the necessity of move limit reduction. However, reduction of the size of the search subregion is essential for optimum designs that are determined by an objective function or constraint curvature which the approximation does not predict. In that case oscillations are usually present, and convergence can only be reached by a repeated move limit reduction.

Convergence is defined by the parameters $vioacc$ and $objacc$, which represent the accuracy of the constraints and the objective function, respectively. The change in objective function value should be smaller than $objacc$, and the constraint violation should not surpass $vioacc$. The same demands are placed on

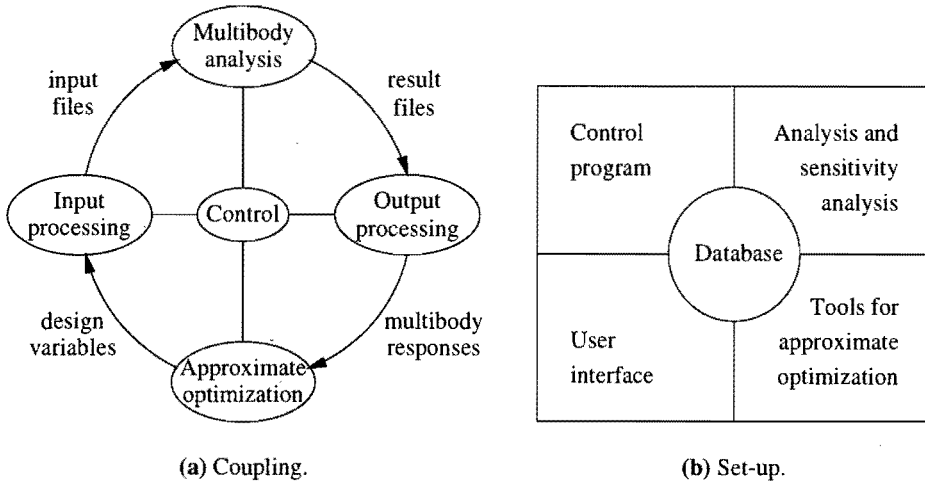


Figure 3.3: Design optimization tool.

the accuracy of the objective function and constraint approximations. For many engineering applications accuracies of about 0.1% will be appropriate.

The complete description of the move limit strategy is given in Appendix A. It is not claimed to be the most efficient strategy that will work for any kind of problem. Probably many improvements are possible. Nevertheless, this simple strategy has worked well for many different design examples. In this thesis the abovementioned parameter settings have been used unless otherwise stated. No tuning has taken place to get the best performance for a specific example.

3.5.5 Aspects of Implementation

The coupling between multibody analysis program and optimization tool is established via the input and output files of the analysis program (see Figure 3.3(a)). The input required for the multibody analysis is parametrized. All items related to the design variables are replaced by unique codes. Then, the input file(s) corresponding to a specific design point can be automatically generated by the optimization tool. The codes are replaced by the parameter values that follow from the design variable values. On return, the multibody program should generate output files containing the calculated multibody responses that contribute to the optimization problem.

Four major parts can be distinguished: the control program, the user interface, the analysis/sensitivity analysis software, and the approximate optimization tools. This is visualized in Figure 3.3(b). In a database all data can be stored necessary for communication. In the current set-up with external analysis package, the control

program is embodied by a Unix-script file. It manages the approximate optimization process. This Unix-script is package dependent, which means that the DADS script (slightly) differs from the MECANO script. The user-interface represents the problem dependent part: parametrized analysis code input files have to be provided, together with Fortran subroutines defining the approximate optimization problem, and optimization input files containing initializations and parameters settings. The approximate optimization tools take care of all tasks related to the optimization, like the generation of the approximate optimization problem and its solution by means of the NAG SQP subroutine (NAG, 1991). They are generally applicable, and do not have to be adjusted for a specific analysis package or design problem.

3.6 Multi-Point Approximations for Noisy Behavior

3.6.1 Approach

For some multibody design problems design sensitivities cannot be accurately calculated. Due to inaccuracies a band of noise may be present on the global response of objective function and constraints. In that case, the optimization tool described in the previous section cannot be applied. Usually, the multibody responses yielding the objective function and constraint values still follow from a computationally expensive numerical analysis. So an alternative optimization strategy should be selected to handle this kind of discontinuously behaving design problems using as least numerical simulations as possible.

A mid-range multi-point approximation concept is able to deal with both the noisy functional behavior and the high computational cost. Several concepts have been published. They all start from the sequential approximate optimization approach. But now just function value data is used to build the approximation models. Vanderplaats' method (1979) adds only one design point to the total set of points during each cycle, and iteratively builds a second order model of the objective function. Free *et al.* (1987) used statistical experimental designs, such as full factorial, screening and central composite designs. Toropov *et al.* (1993, 1996) suggested an experimental design with large difference steps in each design variable direction, either forward or backward depending on the path of optimization.

The mid-range multi-point approach using the experimental design of Toropov *et al.* (1993, 1996) has been implemented*. In each design cycle approximations are built based on the function values calculated at the $n + 1$ points of the experimental

*The multi-point approximate optimization tool was developed during the master's thesis work of Adriaens (1995). The research was carried out in close cooperation with TNO Road-Vehicles Research Institute, Delft (see also Chapter 6).

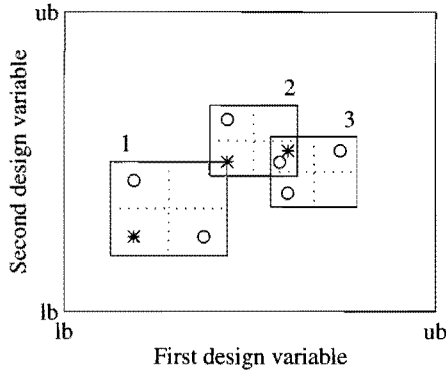


Figure 3.4: A sequence of search subregions and experimental designs for the multi-point approximation concept. Each search subregion has a cycle start design (*) and plan points (o). The dotted lines mark the center of the search subregion. Lower and upper bounds are denoted by lb and ub, respectively.

design (see Figure 3.4). A smooth approximate optimization problem follows that can be solved within the search subregion, separately from the numerical analysis. The position of the approximate optimum (*) with respect to the center of the subregion determines the new search direction and position of the plan points (o) of the following cycle. Basically, the procedure is equal to the optimization process for a local approximation concept outlined in the previous section.

3.6.2 Approximation

Instead of Equations (3.12) and (3.13), the response approximations for the time independent and time dependent case can now be written as:

$$\tilde{r}_h(\mathbf{b}_h^I) = r_h(\mathbf{b}_{0h}^I) + \sum_{k=1}^n (b_{hk}^I - b_{0hk}^I) \frac{r_h(\mathbf{b}_{kh}^I) - r_h(\mathbf{b}_{0h}^I)}{b_{khk}^I - b_{0hk}^I}, \quad (3.23)$$

and:

$$\tilde{r}_h(\mathbf{b}_h^I, t_p) = r_h(\mathbf{b}_{0h}^I, t_p) + \sum_{k=1}^n (b_{hk}^I - b_{0hk}^I) \frac{r_h(\mathbf{b}_{kh}^I, t_p) - r_h(\mathbf{b}_{0h}^I, t_p)}{b_{khk}^I - b_{0hk}^I}, \quad (3.24)$$

respectively. Again, superscript (q) denoting the q -th cycle has been omitted for brevity. Herein, $\mathbf{b}_{0h}^{I(q)}$ is the cycle start design, and \mathbf{b}_{kh}^I ($k = 1, \dots, n$) are the plan points. Data of previous cycles is discarded. This in contrary to Toropov *et al.*

(1993), who used regression techniques to fit their models to most of the available analysis results.

Up to now, in the developed optimization software multi-point approximations can not yet be combined with the introduction of intermediate design variables and intermediate response quantities. So, \mathbf{b}^l equals \mathbf{b} , and both F and \mathbf{g} directly correspond with the responses \mathbf{r} . As a consequence, the optimization process is actually a sequence of linear programming problems, which are solved using the Simplex algorithm of Press *et al.* (1992). However, it is not suggested that intermediate design variables and intermediate response quantities are useless for multi-point approximations. On the contrary, their benefit is independent of the type of approximation concept.

3.6.3 Move Limit Strategy

The move limit strategy mentioned in the previous section can be used for multi-point approximations as well. Instead of Equations (3.20) and (3.21) the search subregion is now defined by:

$$s_i^{l(q)} = b_{0i}^{(q)} - \left(\frac{1 - d_i^{(q)}}{2} + d_i^{(q)} \alpha \right) s_i^{(q)}, \quad i = 1, \dots, n, \quad (3.25)$$

$$s_i^{u(q)} = b_{0i}^{(q)} + \left(\frac{1 + d_i^{(q)}}{2} - d_i^{(q)} \alpha \right) s_i^{(q)}, \quad i = 1, \dots, n, \quad (3.26)$$

where $s_i^{(q)}$ again represents the size of the search subregion. For multi-point approximations, the size of the search subregion has been related to the size of the design space:

$$s_i^{(q)} = m_i^{(q)} (b_i^u - b_i^l), \quad i = 1, \dots, n. \quad (3.27)$$

This implies that both design variable lower and upper bounds have to be present. Parameters $d_i^{(q)} \in \{-1, 1\}$ are the search directions for each design variable, which determine the position of the plan points (\circ) with respect to the cycle start design (\ast) (see Figure 3.4). A direction of 1 or -1 means a difference step forward or backward, respectively. The amount of extrapolation is determined by α . Usually, α is set to a value in between 0.1 and 0.2.

For each new cycle, the move limit strategy has to calculate the corresponding search subregion bounds. In Section 3.5.4 move limit parameters have already been introduced, defining when move limits are reduced or enlarged by means of move limit factors $m_i^{(q)}$. The settings of these parameters are directly influenced by the

accuracy or the noise band of the objective function and constraints. Approximation errors smaller than or equal to variations due to the noise have to be considered as accurate. As a consequence, for increasing noise amplitude, the move limit parameters have to become less strict. The same is true for the convergence criteria. The noise confines the objective function accuracy and maximum constraint violations that can be attained. This means that at the beginning of the optimization, the band of noise has to be estimated. Then, the corresponding parameter settings can be found by adding this percentage of noise to the smooth settings of the previous section.

Additionally to a new size of the search subregion, the move limit strategy has to select a new search direction. The search direction is determined by the position of the approximate optimum design $\mathbf{b}_*^{(q)}$ with respect to the center point $\mathbf{b}_c^{(q)}$ of the search subregion. For the next cycle, parameters $d_i^{(q+1)}$ are calculated from:

$$d_i^{(q+1)} = \frac{b_{*i}^{(q)} - b_{ci}^{(q)}}{|b_{*i}^{(q)} - b_{ci}^{(q)}|}, \quad i = 1, \dots, n, \quad (3.28)$$

with $\mathbf{b}_c^{(q)}$ defined by: $b_{ci}^{(q)} = (s_i^{l(q)} + s_i^{u(q)})/2$. At the start of the optimization, the initial design and move limit factors have to be accompanied by an initial search direction.

3.7 Examples

3.7.1 Impact-Absorber

Max-Value Objective Function with Smooth Functional Behavior

The sequential approximate optimization approach is illustrated for the impact-absorber problem of Section 2.5.1. First of all, consider the case with the maximum acceleration objective function. The optimization is carried out using a local approximation concept, as described in Section 3.5. No intermediate design variables and responses are introduced. As a result, the optimization is a sequential linear programming process. The desired accuracy is set to (cf. Section 2.5.1): $\text{objacc} = 0.001\%$ and $\text{vioacc} = 0.001\%$. The critical constraint bound is set to 0.7, which means that all time point constraints below this value are discarded in the approximate optimization problem.

An optimization is carried out with a constraint regionalization including two additional time points before and after the critical and potentially critical response maxima. Starting from the initial design $b_1 = 1.0$, $b_2 = 0.15$ and the move limit

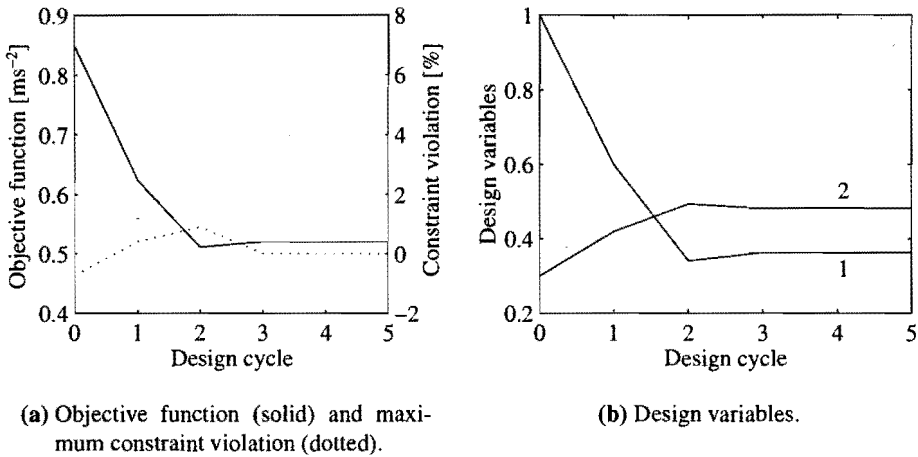


Figure 3.5: Optimization history of the impact-absorber with max-value objective function for two additional time points before and after local maxima.

Cycle	active mvlm	b_1 [Nm ⁻¹]	b_2 [Nsm ⁻¹]	$b_3 = F$ [ms ⁻²]	E_g [%]	V [%]
0 (y)	0	1.000	0.3000	0.8500	0.0	$-7.296 \cdot 10^{-1}$
1 (y)	2	0.6000	0.4200	0.6232	$4.703 \cdot 10^{-1}$	$4.043 \cdot 10^{-1}$
2 (y)	0	0.3411	0.4934	0.5110	$1.625 \cdot 10^{-1}$	$8.996 \cdot 10^{-1}$
3 (y)	0	0.3628	0.4823	0.5203	$3.539 \cdot 10^{-3}$	$5.554 \cdot 10^{-4}$
4 (y)	0	0.3628	0.4823	0.5203	$9.055 \cdot 10^{-9}$	$6.510 \cdot 10^{-9}$
5 (y)	0	0.3628	0.4823	0.5203	0.0	$6.510 \cdot 10^{-9}$

Table 3.1: Optimization history of the impact-absorber with max-value objective function for two additional time points below and above local maxima.

factors $m_1 = m_2 = 0.4$ the optimum design is found after five cycles using 6 function evaluations, and 5 gradient calculations. This is the same optimum design as calculated by the NAG SQP algorithm given in Table 2.1. The optimization history is summarized in Table 3.1 and visualized in Figure 3.5. For each design cycle design variables b_i , objective function F , maximum constraint violation V and maximum approximation error E_g are tabulated. All cycle optima are accepted as starting design of the next cycle, indicated by (y) in the first column. Move limits (two) are active only during the first design cycle. Constraints are included at a maximum of 14 time points (the complete discretization consists of 101 points). The maximum error and constraint violation remain small.

The optimization is restarted, but now just one time point constraint is included

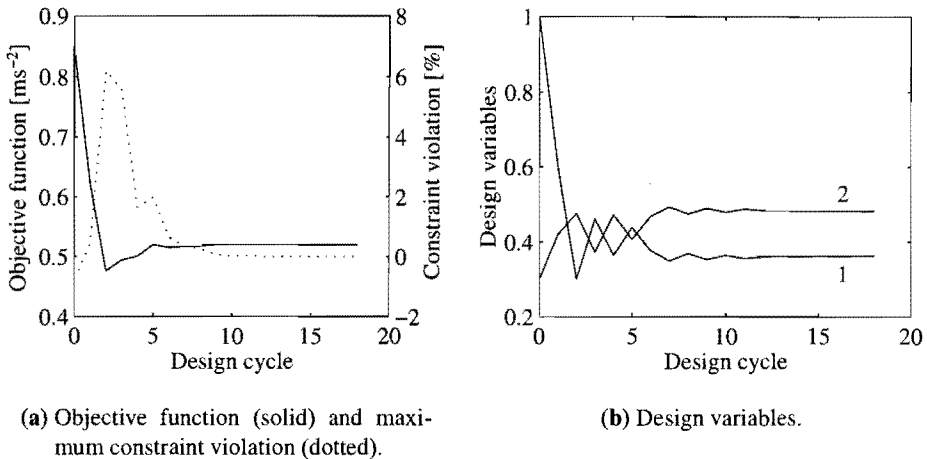


Figure 3.6: Optimization history of the impact-absorber with max-value objective function for zero additional time points before and after local maxima.

for each local maximum. The optimization is visualized in Figure 3.6. Eighteen cycles are necessary to converge. One of the move limits remains active, and convergence can only be reached by a repeated reduction of the search subregion. As a result, an oscillatory behavior of the design variable values occurs. Furthermore, it can be observed that, alike Figure 3.5, the optimum is approached from outside the feasible domain, but now with much larger maximum constraint violations. So, a few more time points for each local maximum yields a much better convergence of the optimization.

Integral Objective Function with Smooth Functional Behavior

The effect of intermediate responses can be illustrated by means of the impact-absorber example with comfort objective function (2.34). The displacement and acceleration responses are linearly approximated with respect to the two design variables, exactly the same as for the max-value example. However, the integral relation between objective function value and acceleration is included in the approximation. The final optimum design is found after 8 cycles using 9 function evaluations and 8 gradient calculations. This is about three times faster compared with the SQP algorithm (see Section 2.5.1). The optimization history is visualized in Figure 3.7, and tabulated in Table 3.2 (the number of active move limits has not been included in this table). During the first four cycles always one move limit is active. Afterwards, automatic convergence occurs without any oscillations. At cy-

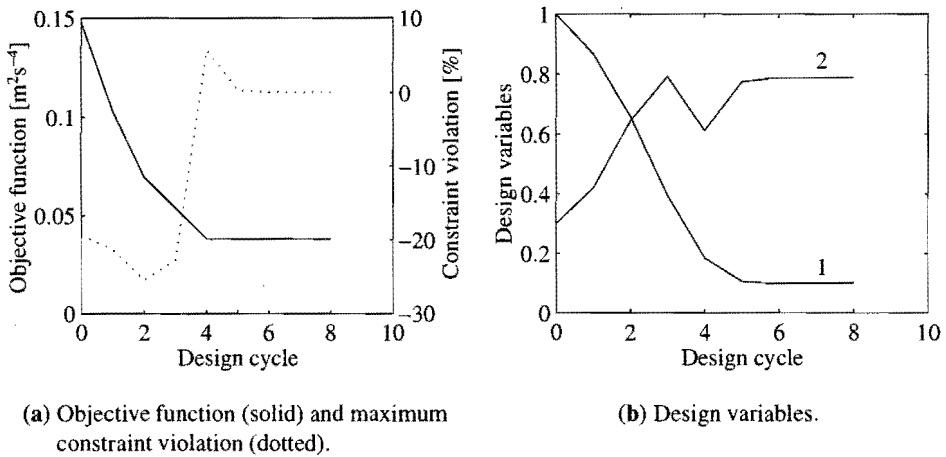


Figure 3.7: Optimization history of the impact-absorber with integral objective function.

Cycle	b_1 [Nm ⁻¹]	b_2 [Nsm ⁻¹]	E_f [%]	E_g [%]	V [%]	F [m ² s ⁻⁴]
0 (y)	1.000	0.3000	0.0	0.0	$-1.938 \cdot 10^1$	0.1480
1 (y)	0.8657	0.4200	$5.901 \cdot 10^{-1}$	$1.095 \cdot 10^{-1}$	$-2.132 \cdot 10^1$	0.1029
2 (y)	0.6596	0.6640	$2.225 \cdot 10^0$	$4.781 \cdot 10^{-1}$	$-2.549 \cdot 10^1$	0.06949
3 (y)	0.3958	0.7933	$3.909 \cdot 10^0$	$2.882 \cdot 10^{-1}$	$-2.277 \cdot 10^1$	0.05384
4 (y)	0.1847	0.6117	$1.352 \cdot 10^1$	$3.801 \cdot 10^0$	$5.663 \cdot 10^0$	0.03805
5 (y)	0.1052	0.7750	$2.494 \cdot 10^{-2}$	$5.378 \cdot 10^{-1}$	$3.192 \cdot 10^{-1}$	0.03793
6 (y)	0.09856	0.7882	$7.365 \cdot 10^{-4}$	$2.286 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	0.03804
7 (y)	0.09850	0.7883	$1.168 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$	$-1.000 \cdot 10^{-7}$	0.03804
8 (y)	0.09850	0.7883	0.0	0.0	$-1.000 \cdot 10^{-7}$	0.03804

Table 3.2: Optimization history of the impact-absorber with integral objective function.

cle four, the maximum approximation error of 13.5% is observed for the objective function, together with the maximum constraint violation of 5.6%.

Max-Value Objective Function with Non-Smooth Functional Behavior[†]

Consider again the impact-absorber optimization problem with max-value objective function, defined by Equations (2.29) to (2.33) with a discretization of the time interval of 0 to 12 s into 101 time points. Suppose the acceleration responses $\ddot{y}(b_1, b_2, t)$ and displacements $y(b_1, b_2, t)$ have a band of artificial 'noise' added to

[†]This subsection has been partly reproduced from Etman *et al.* (1996a) (see also Chapter 6).

the smooth analytical curves $\ddot{y}_a(b_1, b_2, t)$ and $y_a(b_1, b_2, t)$:

$$\ddot{y}(b_1, b_2, t) = \ddot{y}_a(b_1, b_2, t) + 0.0125 \sum_{k=1}^2 \sin(200\pi b_k), \quad (3.29)$$

$$y(b_1, b_2, t) = y_a(b_1, b_2, t) + 0.0125 \sum_{k=1}^2 \sin(200\pi b_k). \quad (3.30)$$

This corresponds with a noise amplitude of about 5 to 10% depending on the place in the design space. The local design sensitivities do not represent the global response behavior anymore. The solution of the noisy optimization problem is a region around the smooth optimum given in Table 3.1.

The multi-point approximate optimization strategy is tested for the following three initial designs: 1) $\mathbf{b}_0^{(1)} = [1, 0.3, 0.843]^T$, 2) $\mathbf{b}_0^{(1)} = [0.7, 0.8, 0.803]^T$, and 3) $\mathbf{b}_0^{(1)} = [0.2, 0.4, 0.409]^T$. The move limit and convergence parameters are set according to an estimated band of noise of 5%. For all initial designs, the start search direction $\mathbf{d}^{(1)}$ and extrapolation factor α are taken $[-1, 1, -1]^T$ and 0.2, respectively. The start dimensions of the search subregion are $\mathbf{s}^{(1)} = [0.3, 0.3, 0.5]^T$. For the third design variable linear functional behavior is present, so there is no need for a limitation of this variable. A value of 0.5 ensures that the corresponding move limit will never become active. Table 3.3 shows the final optimum design found, together with the maximum constraint violation and the number of optimization cycles. For every cycle, one cycle start design and three plan points were analyzed. Three different optima are found in the noise band near the smooth optimum solution.

Initial design	b_1 [Nm ⁻¹]	b_2 [Nsm ⁻¹]	$b_3 = F$ [ms ⁻²]	V [%]	Number of cycles
1	0.385	0.480	0.515	1.28	4 (17)
2	0.356	0.467	0.531	2.21	3 (13)
3	0.365	0.482	0.520	1.05	3 (13)

Table 3.3: Optimum impact-absorber starting from three initial designs. The total number of analyses is given in the last column in between the parentheses.

What happens if the amplitude of the noise is over or under estimated? To answer this question, four different move limit and convergence settings are studied for noise band estimations of 2.5%, 5%, 10% and 20%. The optimization is started from the first initial design. The start search direction and extrapolation factor stay the same. The final optimum designs, the maximum constraint violations and the number of cycles to reach the optimum solution are tabulated in Table 3.4. If a too

small noise band is estimated, bad convergence results. Oscillations occur within or near the optimum region until accidentally a design is found that satisfies the required accuracy. An over estimation, on the other hand, leads to a premature convergence of the optimization process. A reasonable large region in between these extrema is present with a good convergence of the optimization process.

Noise estimation	b_1 [Nm ⁻¹]	b_2 [Nsm ⁻¹]	$b_3 = F$ [ms ⁻²]	V [%]	Number of cycles
2.5%	0.349	0.510	0.516	1.76	10 (41)
5%	0.385	0.480	0.515	1.28	4 (17)
10%	0.364	0.468	0.529	2.74	3 (13)
20%	0.762	0.451	0.684	1.52	1 (5)

Table 3.4: Optimum impact-absorber for four move limit and convergence parameter settings. The total number of analyses is given in the last column in between the parentheses.

3.7.2 Slider-Crank Mechanism

The slider-crank mechanism described in Section 2.5.2 is another example for which the effect of intermediate response variables can be illustrated. Responses x_2 , y_2 and θ_2 are assumed to be not explicitly known, and are linearly approximated with respect to the design variables. Relations (2.35), (2.38) and (2.39) are treated as explicitly known, and included in the approximate optimization problem. The desired final accuracy is again $\text{objacc} = 0.001\%$ and $\text{vioacc} = 0.001\%$. A sequence of ten approximate nonlinear programming problems yields the final optimum solution, which exactly corresponds with the solution obtained by the NAG SQP algorithm. However, nearly five times less function value and design sensitivity evaluations were required.

The optimization history is given in Table 3.5 and visualized in Figure 3.8. The first design cycle is repeated (with smaller move limits) since the approximation error for initial move limit factors of $m_i = 0.4$ ($i = 1, \dots, 8$) is far too large. Afterwards a good convergence occurs with a steady reduction of the number of active move limits. During the last three cycles no move limits are active anymore and the maximum design variable change $\max \Delta b_i$ shows an exponential decrease. Constraint violations do not occur since the movability constraints are explicitly known and included in the approximate optimization problem. If the optimization is started with move limit factors $m_i = 0.3$, just eight cycles will be necessary to find the optimum solution. Then, a much smaller approximation error occurs at the first design cycle.

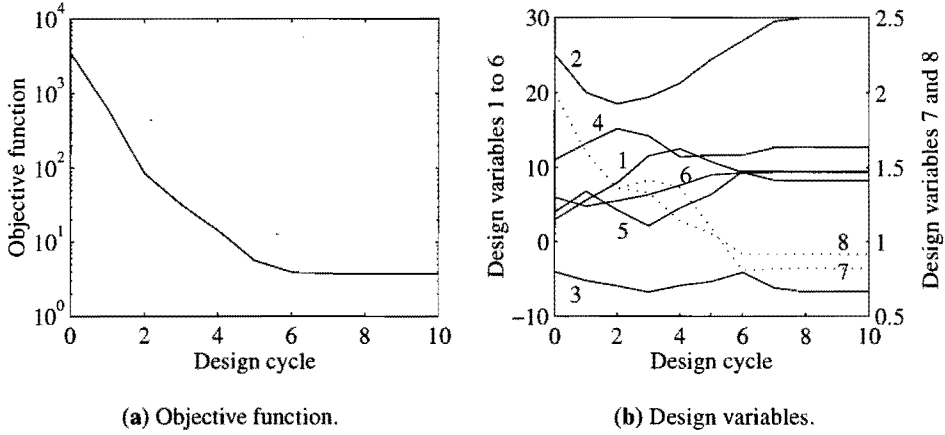


Figure 3.8: Slider-crank mechanism.

Cycle	active mvlm	$\max \Delta b_i$ [%]	E_f [%]	F
0 (y)	0	$0.000 \cdot 10^0$	$0.000 \cdot 10^0$	$3.486 \cdot 10^3$
1 (n)	5	$4.000 \cdot 10^1$	$8.541 \cdot 10^1$	$1.303 \cdot 10^2$
1 (y)	8	$2.000 \cdot 10^1$	$2.870 \cdot 10^1$	$6.260 \cdot 10^2$
2 (y)	7	$1.500 \cdot 10^1$	$2.506 \cdot 10^0$	$8.500 \cdot 10^1$
3 (y)	4	$2.000 \cdot 10^1$	$1.691 \cdot 10^{-1}$	$3.225 \cdot 10^1$
4 (y)	4	$2.667 \cdot 10^1$	$6.177 \cdot 10^0$	$1.455 \cdot 10^1$
5 (y)	2	$2.000 \cdot 10^1$	$5.953 \cdot 10^0$	$5.658 \cdot 10^0$
6 (y)	1	$2.667 \cdot 10^1$	$2.263 \cdot 10^0$	$3.942 \cdot 10^0$
7 (y)	1	$3.556 \cdot 10^1$	$4.954 \cdot 10^{-1}$	$3.772 \cdot 10^0$
8 (y)	0	$1.288 \cdot 10^1$	$9.316 \cdot 10^{-3}$	$3.746 \cdot 10^0$
9 (y)	0	$1.085 \cdot 10^0$	$1.353 \cdot 10^{-4}$	$3.746 \cdot 10^0$
10 (y)	0	$3.248 \cdot 10^{-2}$	$4.511 \cdot 10^{-7}$	$3.746 \cdot 10^0$

Table 3.5: Optimization history of the slider-crank mechanism.

3.8 Conclusion and Discussion

Two types of approximation concepts have been proposed for the optimization of multibody systems: single-point local and multi-point mid-range. The former is especially suitable for smooth optimization problems. The latter will prove its value if some kind of noise is present and design sensitivities are of no use. For both concepts, the sequential approximate optimization process is basically the same. With slightly different convergence parameter values, the same move limit strategy can be applied. In both cases, constraint deletion can reduce the approximate optimization problem towards manageable proportions. For the local concept the computational cost can be reduced if the sensitivity analysis is able to calculate just the gradients of the retained constraints. In the following chapter more attention is paid to this subject.

Intermediate design variables and responses can highly improve the approximations, and thus enhance the optimization process. An important precondition is that appropriate intermediate relations are selected, otherwise bad convergence may occur. The examples show that significant improvements can be found compared with a direct coupling with the SQP-algorithm: the number of multibody analyses and design sensitivity analyses are significantly smaller. Instead, the objective function and constraint approximations are quite frequently evaluated. So, the actual savings in computer time are not so much present for the analytical examples, but become much more apparent for numerically expensive multibody analyses.

To apply the multi-point approach in the case of noisy functional behavior, one has to have an idea of the amplitude of the noise. Engineering experience is required to properly select the move limit parameters and convergence criteria. In the ideal situation, the optimization software would automatically estimate the noise amplitudes of all objective function and constraints, and re-tune the settings accordingly. To achieve this, one can for example try to use the analysis data of previous cycles. However, it will be difficult to distinguish the noise from the global functional behavior. An other option is to enlarge the number of plan points in the experimental design, for example two or three times the number of model parameters. Sufficient data is then available to locally estimate the noise amplitudes. But then the increase of the number of numerical analyses is a serious drawback. Without question, it is preferable to know the severity of the noise before the start of the optimization, but unfortunately this is often too high a demand.

Chapter 4

Design Sensitivity Analysis

4.1 Introduction

Local approximation concepts require gradient information. For each new approximate optimization cycle a design sensitivity analysis has to provide the derivatives of the multibody responses with respect to the design variables. Actually, the design sensitivity analysis is an extension of the multibody analysis. It may be preceded by a constraint screening to mark the important constraints in the approximate optimization problem. Then, gradients need to be calculated just for the responses related to the critical and potentially critical constraints. This may help to reduce the cost of the design sensitivity analysis. Efficiency together with accuracy are the most important factors determining the effectiveness of design sensitivities in numerical optimization.

A finite difference sensitivity analysis is a very simple and straightforward way to obtain design sensitivities. In every design variable direction at least one additional slightly perturbed design point is analyzed, from which derivatives can be calculated. Well-known concepts are forward, backward, and central difference schemes. Due to the additional set of numerical analyses, finite differencing is quite numerically expensive, and does not benefit from the constraint screening. Furthermore, it may suffer from numerical inaccuracies. Nevertheless, it is often used, not just because of its simplicity, but usually because there is no alternative at all. Think for example of coupling the optimization tool described in Section 3.5 with an external multibody code such as DADS (CADSi, 1995) or MECANO (Samtech, 1994).

In the ideal case, the multibody analysis software is able to calculate the design sensitivities corresponding to the multibody responses. For this purpose, the soft-

Reference	Coordinate formulation	Type of equations	Generation equations	Method
Haug and Arora (1979)	-	AE,ODE	-	D,A
Sohoni and Haug (1982)	C	AE	N	A
Haug <i>et al.</i> (1984)	C	DAE	N	A
Hsieh and Arora (1984)	-	ODE	-	D,A
Chang and Nikravesh (1985)	-	DAE	-	D
Haug (1987)	-,C	ODE,DAE	-,N	D,A
Tak and Kim (1989)	R	DAE	N	D
Ashrafioun and Mani (1990)	R	DAE	S	D
Bestle and Eberhard (1992)	R	ODE	S	A
Bestle and Seybold (1992)	-	DAE	-	A
Bestle (1994)	R	ODE,DAE	S	D,A
Dias and Pereira (1994)	C	DAE	S	D
Eberhard (1996)	R	ODE	S,AD	A
Fleury (1996)	C	DAE	S	D,A
Hansen and Tortorelli (1996)	R	AE	-	D
Pagalday <i>et al.</i> (1996)	-	DAE	S	D
Pesch <i>et al.</i> (1996)	C	AE	S	D

Table 4.1: An overview of some relevant references on design sensitivity analysis of multi-body systems. For each reference four aspects are identified. The coordinate formulation from which the governing equations are generated can be either relative (R) or Cartesian (C). Furthermore, three types of equations are distinguished: algebraic (AE), ordinary differential (ODE), and differential-algebraic (DAE). The sensitivity equations can be developed numerically (N), symbolically (S), or by means of automatic differentiation (AD). Finally, the design sensitivity method is either direct (D) or adjoint (A). An empty entry (-) means that no specific choice was made with respect to this argument.

ware should also generate and solve the governing equations of the design sensitivities. A substantial amount of literature is available on this subject. In Table 4.1 an overview is given of some relevant references in order of year of publication. The governing equations can be generated numerically or symbolically, based on a relative or Cartesian coordinate formulation. Relative coordinates yield a minimal set of equations, and are numerically more efficient than the Cartesian formulation that results in a much larger set of equations. However, the latter approach benefits from the easy and general development of the equations of motion.

Two different methods can be identified to analytically obtain the design sensitivity equations: the direct and the adjoint approach. The direct method differentiates the governing equations of motion with respect to the design variables. As a consequence, for each design variable a new set of equations has to be solved.

When there are many design variables, and sensitivities are required for only a few responses or constraints, the adjoint method may be numerically more efficient. The adjoint technique introduces so-called adjoint variables and rewrites the sensitivity equations of the direct method such that equations follow for each response or constraint function. This means that the number of adjoint equations does not depend on the number of design variables anymore.

The adjoint variable method was introduced for dynamic mechanical systems by Haug and Arora (1979), as an alternative for direct differentiation. In the years thereafter, it became the preferred approach for the design sensitivity analysis of multibody systems. Firstly, pure kinematically driven systems were considered (Sohoni and Haug, 1982), followed by constrained dynamic mechanical systems (Haug *et al.*, 1984; Haug, 1987). Chang and Nikravesh (1985) switched back to the direct method because of numerical and implementational difficulties in the adjoint method. Tak and Kim (1989) also reconsidered the preference for the adjoint method, and selected the direct method due to newly emerging parallel computing technologies. The nineties are characterized by the advent of symbolic computing to obtain the design sensitivity equations of motion, using either the direct method (Ashrafioun and Mani, 1990) or the adjoint method (Bestle and Eberhard, 1992; Bestle and Seybold, 1992). Besides, the design sensitivity analysis of mechanism kinematics regained new interest (e.g., Hansen and Tortorelli, 1996; Pesch *et al.*, 1996).

Recently, Eberhard (1996) proposed to use automatic differentiation for computation of the design sensitivity equations. Automatic differentiation (AD) is a methodology for developing sensitivity-enhanced versions of arbitrary computer programs (Bischof, 1996). It applies the chain rule of differentiation between user-defined dependent and independent variables in the program, and automatically generates new program text to calculate the derivatives of the dependent variables with respect to the independent variables. To this end, the source code of the program should be available (in Fortran or C). Automatic differentiation produces new code that computes the derivatives faster and more accurately (to machine precision) as compared with finite differencing, provided that the original program is not too large (Barthelemy and Hall, 1995). It is able to deal with multibody problems that are difficult to solve by symbolic computation.

This chapter starts with a short description of finite difference sensitivity analysis, and continues with the analytical derivation of the design sensitivity equations. The text reflects the present state-of-the-art as can be found in the references of Table 4.1. Automatic differentiation is not described in further detail. For the analytical sensitivities, three types of equations are considered: algebraic, ordinary differential and differential-algebraic equations. Each of them is treated in

a separate section. Examples are given by means of the impact-absorber and the slider-crank mechanism of Section 2.5. Finally, this chapter is concluded with a discussion. Herein, the consequences and conditions of the proposed approximate optimization strategy for the design sensitivity analysis are of main interest.

4.2 Finite Differences

An easy way to approximate the derivative of a function $f(b)$ is to use the forward difference formula. Then, $\frac{df}{db}$ is approximated by:

$$\frac{\Delta f}{\Delta b} = \frac{f(b + \Delta b) - f(b)}{\Delta b}. \quad (4.1)$$

The problem is now to select a proper step size $\Delta b > 0$. For large values of Δb approximation errors arise caused by the neglected terms of the Taylor series:

$$f(b + \Delta b) = f(b) + \Delta b \frac{df}{db}(b) + \frac{(\Delta b)^2}{2} \frac{d^2 f}{db^2}(b) + \dots \quad (4.2)$$

From Equations (4.1) and (4.2) it follows that this truncation error is of the order:

$$e_t \sim \Delta b \left| \frac{d^2 f}{db^2}(b) \right|. \quad (4.3)$$

On the other hand, inaccuracies in the computed function values become important for small step sizes, yielding a condition error of the order:

$$e_c \sim \frac{|f(b)|}{\Delta b} \epsilon_f, \quad (4.4)$$

with ϵ_f the relative accuracy of $f(b)$. For multibody systems ϵ_f is directly related to the accuracy used to solve the governing equations.

The optimal step size follows from the minimization of the sum of the truncation and condition error:

$$\Delta b_{opt} \sim b_s \sqrt{\epsilon_f}, \quad (4.5)$$

where b_s is the curvature scale or characteristic scale of function f :

$$b_s = \sqrt{\left| f(b) / \frac{d^2 f}{db^2}(b) \right|}. \quad (4.6)$$

Using this step size, and assuming that f , $\frac{df}{db}$ and $\frac{d^2f}{db^2}$ have the same order of magnitude, the relative error in the approximation of the derivative $(e_t + e_c)/\frac{df}{db}$ is $\sqrt{\epsilon_f}$. In absence of any other information, Press *et al.* (1992) suggest $b_s = b$, except if b comes near zero.

To improve the accuracy, the central difference formula can be used:

$$\frac{\Delta f}{\Delta b} = \frac{f(b + \Delta b) - f(b - \Delta b)}{2\Delta b}. \quad (4.7)$$

The truncation error is then:

$$e_t \sim (\Delta b)^2 \left| \frac{d^3f}{db^3}(b) \right|. \quad (4.8)$$

The condition error is about the same as compared with forward difference. As a result, the optimal choice of Δb becomes:

$$\Delta b_{opt} \sim b_s \sqrt[3]{\epsilon_f}, \quad (4.9)$$

with a maximum accuracy of $\epsilon_f^{2/3}$. If $\epsilon_f = 10^{-6}$, Equation (4.7) is an order of magnitude better than Equation (4.1).

In Chapters 5 and 7 the forward difference formula is applied to obtain the derivatives of the multibody responses calculated by MECANO and DADS with respect to the design variables. According to Equation (4.5), finite difference step sizes are calculated from:

$$\Delta b = (1 + |b|)\beta. \quad (4.10)$$

Herein, $(1 + |b|)$ represents the curvature scale b_s . Parameter β is the relative difference step, and is usually set to the square root of the integration accuracy. Though, it never exceeds 10^{-3} since the output-files of both MECANO and DADS do not contain more than six significant digits. For time dependent responses it is important that the time points of the perturbed and unperturbed response values match. Using MECANO and DADS, this precondition is met by computing the multibody responses at a predefined grid of time points.

4.3 Kinematically Driven Multibody Systems

4.3.1 Algebraic Equations of Motion

Consider a multibody system that is made up of rigid bodies with holonomic couplings. Holonomic couplings constrain the relative position between pairs of bodies. If driving conditions are specified for the remaining degrees of freedom, then

the configuration of the system can be determined as a function of time. The generalized coordinates $\mathbf{q}(t)$ have to be solved from a set of nonlinear algebraic equations*:

$$\Phi(\mathbf{q}, \mathbf{b}, t) = \mathbf{0}, \quad t \in [t_0, t_f], \quad (4.11)$$

given predefined design variable values \mathbf{b} . The Jacobian matrix $\Phi_{\mathbf{q}}$ plays a central role in the iterative solution of Equation (4.11). Often a Newton-Raphson algorithm is used that is quadratically convergent, provided that the Jacobian is nonsingular and the algorithm converges. These preconditions may not be met, for example if the system has yet to be assembled. For this reason Haug (1989) calculates an initial configuration $\mathbf{q}(t_0)$ by employing an optimization algorithm that minimizes the residuals present in Equation (4.11) towards zero.

The corresponding velocities and accelerations are obtained by differentiating Equation (4.11) with respect to time using the chain rule of differentiation. The velocity and acceleration equations are (Haug, 1989):

$$\Phi_{\mathbf{q}}\dot{\mathbf{q}} = -\Phi_t \equiv \boldsymbol{\nu}, \quad (4.12)$$

$$\Phi_{\mathbf{q}}\ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\Phi_{\mathbf{q}t}\dot{\mathbf{q}} - \Phi_{tt} \equiv \boldsymbol{\gamma}. \quad (4.13)$$

Subscript \mathbf{q} and t denote partial differentiation with respect to \mathbf{q} and t , respectively. A full description of the matrix notation employed here is given on page xiii. The Jacobian matrix $\Phi_{\mathbf{q}}$, and the right sides $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$ can be numerically assembled from a library of joint equations, or generated by means of symbolic computation. When the accelerations are available, forces imposed on the joints can be calculated from an inverse dynamic analysis using:

$$\Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}^A - \mathbf{M}\ddot{\mathbf{q}}. \quad (4.14)$$

Herein, $\mathbf{M}(\mathbf{q}, \mathbf{b})$ is the mass matrix, and $\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t)$ are the externally applied forces. The constraint reaction forces in the system are uniquely determined by the Lagrange multipliers $\boldsymbol{\lambda}$ (Haug, 1989).

Several steps in the numerical solution of the kinematic equations can be distinguished. Firstly, the multibody system has to be assembled, starting from a (poor) user-defined guess of the initial body positions. In some cases this may be physically impossible. Once an assembled configuration is found, the kinematic position equations (4.11) can be solved for a grid of time points, as long as the determinant

*The algebraic equations due to kinematic couplings are often called multibody constraints or just constraints. Since they are not included as constraints in the optimization problem this term is not used here (see Chapter 2).

of the Jacobian stays non-zero. The iterative solution process for a time point t_p can benefit from calculations at the previous time instant t_{p-1} . A rapid convergence may be expected if the increments in time points are not too large. Next, velocities $\dot{\mathbf{q}}(t_p)$, accelerations $\ddot{\mathbf{q}}(t_p)$, and Lagrange multipliers $\boldsymbol{\lambda}(t_p)$ can be easily obtained from Equations (4.12), (4.13) and (4.14) by linear matrix algebra. Here the decomposition of the Jacobian matrix $\Phi_{\mathbf{q}}(t_p)$ can be re-used.

4.3.2 Algebraic Sensitivity Equations

Design sensitivities can be analytically derived by differentiation of the kinematic equations introduced in the previous section. Differentiating the kinematic position equations (4.11) with respect to design variable b ($b \in \mathbf{b}$) yields:

$$\Phi_{\mathbf{q}} \frac{d\mathbf{q}}{db} = -\Phi_b. \quad (4.15)$$

The derivatives at a certain time instant t_p can be obtained by simply solving this set of linear algebraic equations for each design variable. The decomposition of the Jacobian matrix is still available. Only one additional term Φ_b has to be developed. This is called the direct differentiation approach.

In many cases one does not explicitly need the complete column of derivatives $\frac{d\mathbf{q}}{db}$. For example, the constraint screening may show that only one of the generalized coordinates is important. To this end, a response r may be defined that linearly depends on \mathbf{q} :

$$r = \mathbf{w}^T \mathbf{q}. \quad (4.16)$$

One can easily pick one element out of \mathbf{q} by setting the corresponding element in \mathbf{w} to a value of one, and all other elements to zero. The adjoint variable approach uses relation (4.16) to avoid the calculation of $\frac{d\mathbf{q}}{db}$ for each design variable.

The adjoint sensitivity equations of the kinematic positions are derived as follows. Starting point is that the design sensitivities of response r are needed instead of \mathbf{q} :

$$\frac{dr}{db} = \mathbf{w}^T \frac{d\mathbf{q}}{db}. \quad (4.17)$$

Add to (4.17) the derivatives of the kinematic position equations (4.15) multiplied by adjoint variables[†] $\boldsymbol{\xi}(t)$ (Haftka and Gürdal, 1992):

$$\frac{dr}{db} = \mathbf{w}^T \frac{d\mathbf{q}}{db} + \boldsymbol{\xi}^T \left(-\Phi_{\mathbf{q}} \frac{d\mathbf{q}}{db} - \Phi_b \right). \quad (4.18)$$

[†]Haftka and Gürdal (1992) call them Lagrange multipliers, but to avoid confusion with the Lagrange multipliers $\boldsymbol{\lambda}$ in Equation (4.14) solely the term adjoint variables is used.

This addition has no influence on the value of $\frac{dr}{db}$ since the expression between brackets is actually zero. As a consequence, any column ξ may be selected. Rearranging Equation (4.18) gives:

$$\frac{dr}{db} = (\mathbf{w}^T - \xi^T \Phi_q) \frac{dq}{db} - \xi^T \Phi_b. \quad (4.19)$$

We do not want to calculate $\frac{dq}{db}$, and therefore choose the adjoint variables ξ such that:

$$\mathbf{w}^T - \xi^T \Phi_q = \mathbf{0}^T, \quad (4.20)$$

which means that the adjoint equations become:

$$\Phi_q^T \xi = \mathbf{w}, \quad (4.21)$$

and the sensitivity of the response:

$$\frac{dr}{db} = -\xi^T \Phi_b. \quad (4.22)$$

The column of adjoint variables ξ for a time point can be calculated independent of the number of the design variables. The main computational cost is now determined by the number of responses r sensitivities are required for.

Sensitivity equations of the velocities, accelerations, and the Lagrange multipliers of the constraint reaction forces can be derived in a similar manner. Differentiation of the kinematic velocity equations (4.12):

$$(\Phi_q \dot{q})_q \frac{dq}{db} + (\Phi_q \dot{q})_{\dot{q}} \frac{d\dot{q}}{db} + (\Phi_q \dot{q})_b = \nu_q \frac{dq}{db} + \nu_b \quad (4.23)$$

yields the velocity sensitivity equations:

$$\Phi_q \frac{d\dot{q}}{db} = -[(\Phi_q \dot{q})_q - \nu_q] \frac{dq}{db} - (\Phi_{qb} \dot{q} - \nu_b). \quad (4.24)$$

Moreover, the following expressions for $\frac{d\ddot{q}}{db}$ and $\frac{d\lambda}{db}$ are obtained:

$$\Phi_q \frac{d\ddot{q}}{db} = -[(\Phi_q \ddot{q})_q - \gamma_q] \frac{dq}{db} + \gamma_{\dot{q}} \frac{d\dot{q}}{db} - (\Phi_{qb} \ddot{q} - \gamma_b), \quad (4.25)$$

$$\begin{aligned} \Phi_q^T \frac{d\lambda}{db} = & -[(\mathbf{M}\ddot{q})_q + (\Phi_q^T \lambda)_q - \mathbf{Q}_q^A] \frac{dq}{db} + \mathbf{Q}_{\dot{q}}^A \frac{d\dot{q}}{db} - \mathbf{M} \frac{d\ddot{q}}{db} \\ & - (\mathbf{M}_b \ddot{q} + \Phi_{qb}^T \lambda - \mathbf{Q}_b^A). \end{aligned} \quad (4.26)$$

Although these expressions are slightly more complex, the direct and adjoint approach are equivalent to the procedure outlined for Equation (4.15). The direct approach requires the right hand sides to be evaluated, and the solution of the linear set of equations for each design variable. The adjoint method first rewrites these equations before sensitivities are calculated.

The adjoint variable approach is demonstrated using a general expression of response r :

$$r = r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \mathbf{b}, t). \quad (4.27)$$

Design sensitivities of r are desired:

$$\frac{dr}{db} = r_{\mathbf{q}} \frac{d\mathbf{q}}{db} + r_{\dot{\mathbf{q}}} \frac{d\dot{\mathbf{q}}}{db} + r_{\ddot{\mathbf{q}}} \frac{d\ddot{\mathbf{q}}}{db} + r_{\boldsymbol{\lambda}} \frac{d\boldsymbol{\lambda}}{db} + r_b, \quad (4.28)$$

without explicitly calculating the position, velocity and acceleration sensitivities $\frac{d\mathbf{q}}{db}$, $\frac{d\dot{\mathbf{q}}}{db}$ and $\frac{d\ddot{\mathbf{q}}}{db}$, and the sensitivities of the Lagrange multipliers $\frac{d\boldsymbol{\lambda}}{db}$. Therefore, multiply Equations (4.15), (4.24), (4.25), and (4.26) by adjoint variables $\boldsymbol{\xi}(t)$, $\boldsymbol{\zeta}(t)$, $\boldsymbol{\eta}(t)$, and $\boldsymbol{\mu}(t)$, respectively. Add these expressions to (4.28), rearrange, and select the adjoint variables to remove terms with $\frac{d\mathbf{q}}{db}$, $\frac{d\dot{\mathbf{q}}}{db}$, $\frac{d\ddot{\mathbf{q}}}{db}$ and $\frac{d\boldsymbol{\lambda}}{db}$. Then, assuming symmetry of the mass matrix \mathbf{M} , the following adjoint equations are obtained:

$$\begin{aligned} \boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\xi} = & r_{\mathbf{q}}^T - [(\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} - \boldsymbol{\nu}_{\mathbf{q}}]^T \boldsymbol{\zeta} - [(\boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}})_{\mathbf{q}} - \boldsymbol{\gamma}_{\mathbf{q}}]^T \boldsymbol{\eta} \\ & - [(\mathbf{M} \ddot{\mathbf{q}})_{\mathbf{q}} + (\boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda})_{\mathbf{q}} - \mathbf{Q}_{\mathbf{q}}^A]^T \boldsymbol{\mu}, \end{aligned} \quad (4.29)$$

$$\boldsymbol{\Phi}_{\dot{\mathbf{q}}}^T \boldsymbol{\zeta} = r_{\dot{\mathbf{q}}}^T + \boldsymbol{\gamma}_{\dot{\mathbf{q}}}^T \boldsymbol{\eta} + \mathbf{Q}_{\dot{\mathbf{q}}}^A \boldsymbol{\mu}, \quad (4.30)$$

$$\boldsymbol{\Phi}_{\ddot{\mathbf{q}}}^T \boldsymbol{\eta} = r_{\ddot{\mathbf{q}}}^T - \mathbf{M} \boldsymbol{\mu}, \quad (4.31)$$

$$\boldsymbol{\Phi}_{\boldsymbol{\lambda}} \boldsymbol{\mu} = r_{\boldsymbol{\lambda}}^T, \quad (4.32)$$

with:

$$\begin{aligned} \frac{dr}{db} = & r_b - \boldsymbol{\xi}^T \boldsymbol{\Phi}_b - \boldsymbol{\zeta}^T (\boldsymbol{\Phi}_{\mathbf{q}b} \dot{\mathbf{q}} - \boldsymbol{\nu}_b) - \boldsymbol{\eta}^T (\boldsymbol{\Phi}_{\mathbf{q}b} \ddot{\mathbf{q}} - \boldsymbol{\gamma}_b) \\ & - \boldsymbol{\mu}^T (\mathbf{M}_b \ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}b}^T \boldsymbol{\lambda} - \mathbf{Q}_b^A). \end{aligned} \quad (4.33)$$

The numerical solution comprises of consecutively solving $\boldsymbol{\mu}$, $\boldsymbol{\eta}$, $\boldsymbol{\zeta}$, and $\boldsymbol{\xi}$ from (4.32), (4.31), (4.30), and (4.29), respectively. This corresponds with the procedure proposed by Sohoni and Haug (1982). Furthermore, note that response equation (4.27) includes linear relation (4.16) by selecting $r_{\mathbf{q}} = \mathbf{w}^T$, and $r_{\dot{\mathbf{q}}} = r_{\ddot{\mathbf{q}}} = r_{\boldsymbol{\lambda}} = \mathbf{0}^T$. Then Equations (4.29) to (4.32) reduce to Equation (4.21), and Equation (4.33) becomes (4.22).

4.3.3 Discussion

The sensitivity equations of kinematically driven multibody systems resemble the velocity and acceleration equations. They are all linear and can only be consecutively solved. When derivatives of e.g. the acceleration are desired, quite a lot of computations need to be done beforehand. This applies for both the direct and the adjoint technique. However, the direct sensitivity equations are decoupled for each design variable, whereas the adjoint sensitivity equations are decoupled for each response function. So, multiple computer processors may be effectively applied during the multibody design sensitivity analysis.

The Jacobian matrix plays an important role in the kinematic analysis of multibody systems. Its decomposition is required to compute the positions, velocities, accelerations and Lagrange multipliers, as well as all corresponding design sensitivities. However, for lock-up and bifurcation the Jacobian becomes singular. In the case of lock-up the mechanism cannot move any further, whereas for bifurcation a branching to two possible paths of motion is present. As a consequence, computational difficulties arise near such singular configurations. Therefore, constraints are desired that can avoid lock-up and bifurcation, but it is difficult to mathematically formulate them. The problem is that the only warning of a possible singularity nearby is a bad condition number of the Jacobian matrix or high velocities and accelerations. To make matters worse, violation of a singular configuration constraint may cause the numerical analysis to break down due to lock-up, or the bifurcation to take place.

4.4 Multibody Systems described by Ordinary Differential Equations

4.4.1 Ordinary Differential Equations of Motion

The dynamic behavior of a multibody system with tree structure (no closed loops) can be described by a set of ordinary differential equations. A relative coordinate formulation is applied such that no algebraic equations due to kinematic joints arise:

$$\begin{aligned} \mathbf{M}(\mathbf{q}, \mathbf{b})\ddot{\mathbf{q}} &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t), & \mathbf{q}(t_0) &= \mathbf{q}_0, \\ & & \dot{\mathbf{q}}(t_0) &= \dot{\mathbf{q}}_0. \end{aligned} \quad (4.34)$$

Herein, mass matrix \mathbf{M} , generalized applied force \mathbf{Q}^A , and initial conditions $\mathbf{q}(t_0)$ and $\dot{\mathbf{q}}(t_0)$ can be identified. The column of generalized forces includes externally applied forces and torques, as well as forces due to Coriolis effects. For simplicity, constant initial conditions are considered, although for some applications they

may explicitly or implicitly depend on the design variables. This set of second-order equations can be rewritten in first-order form, which yields for holonomic kinematic couplings:

$$\dot{\mathbf{y}} = \mathbf{z}, \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad (4.35)$$

$$\mathbf{M}(\mathbf{y}, \mathbf{b})\dot{\mathbf{z}} = \mathbf{Q}^A(\mathbf{y}, \mathbf{z}, \mathbf{b}, t), \quad \mathbf{z}(t_0) = \mathbf{z}_0. \quad (4.36)$$

The new generalized positions \mathbf{y} and velocities \mathbf{z} are equal to \mathbf{q} and $\dot{\mathbf{q}}$, respectively.

Well-established methods are available to numerically integrate first-order initial value problems. They all boil down to the evaluation of the derivatives $\dot{\mathbf{y}}$ and $\dot{\mathbf{z}}$ at several time instants. The propagation of \mathbf{y} and \mathbf{z} in time is computed by adding increments corresponding to the derivatives multiplied by step sizes (Press *et al.*, 1992). Usually, adaptive step size control is applied. In Equation (4.36) calculation of $\dot{\mathbf{z}}$ requires matrix \mathbf{M} and right-hand side \mathbf{Q}^A to be evaluated, followed by the solution of the linear algebraic equations. For multibody systems this may be rather expensive. Therefore, the integration method should be efficient as well as accurate.

4.4.2 First-Order Differential Sensitivity Equations

Equations (4.35) and (4.36) are differentiated with respect to design variable b to obtain:

$$\frac{d\dot{\mathbf{y}}}{db} = \frac{d\mathbf{z}}{db}, \quad (4.37)$$

$$\mathbf{M} \frac{d\dot{\mathbf{z}}}{db} = -[(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A] \frac{d\mathbf{y}}{db} + \mathbf{Q}_z^A \frac{d\mathbf{z}}{db} - (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A), \quad (4.38)$$

with initial conditions:

$$\frac{d\mathbf{y}}{db}(t_0) = \mathbf{0}, \quad (4.39)$$

$$\frac{d\mathbf{z}}{db}(t_0) = \mathbf{0}. \quad (4.40)$$

The direct method solves $\frac{d\mathbf{y}}{db}(t)$ and $\frac{d\mathbf{z}}{db}(t)$ for each design variable $b \in \mathbf{b}$. Remark that, in contrary to Equation (4.36), sensitivity equation (4.38) is linear in the unknowns $\frac{d\mathbf{y}}{db}$ and $\frac{d\mathbf{z}}{db}$.

The adjoint variable approach starts from a response \bar{r} that is some function of the generalized positions, velocities and accelerations. Consider an integral function of the form:

$$\bar{r} = \int_{t_0}^{t_f} p(\mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{b}, t) dt, \quad (4.41)$$

which covers a broad class of functions. It includes, for example, responses at a one specific time point t_p by:

$$p(\mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{b}, t) = r(\mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{b}, t)\delta(t - t_p), \quad (4.42)$$

with $\delta(t - t_p)$ a Dirac delta function. Differentiation of Equation (4.41) gives:

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} \left(p_y \frac{dy}{db} + p_z \frac{dz}{db} + p_{\dot{z}} \frac{d\dot{z}}{db} + p_b \right) dt. \quad (4.43)$$

If t_0 and t_f depend on the design variables, the Leibniz rule of differentiation has to be applied (see e.g. Abramowitz and Stegun, 1965), which means that two additional terms pop up.

The adjoint method avoids to explicitly calculate sensitivities $\frac{dy}{db}$, $\frac{dz}{db}$, and $\frac{d\dot{z}}{db}$. To accomplish this, one may apply the following procedure. Firstly, multiply Equation (4.37) by adjoint variables $\boldsymbol{\mu}(t)$, and integrate by parts:

$$\int_{t_0}^{t_f} \boldsymbol{\mu}^T \left(-\frac{d\dot{\mathbf{y}}}{db} + \frac{d\mathbf{z}}{db} \right) dt = - \left[\boldsymbol{\mu}^T \frac{d\dot{\mathbf{y}}}{db} \right]_{t_0}^{t_f} + \int_{t_0}^{t_f} \dot{\boldsymbol{\mu}}^T \frac{d\dot{\mathbf{y}}}{db} dt + \int_{t_0}^{t_f} \boldsymbol{\mu}^T \frac{d\mathbf{z}}{db} dt. \quad (4.44)$$

Do the same for Equation (4.38) using adjoint variables $\boldsymbol{\zeta}(t)$:

$$\begin{aligned} & \int_{t_0}^{t_f} \boldsymbol{\zeta}^T \left\{ -\mathbf{M} \frac{d\dot{\mathbf{z}}}{db} - [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A] \frac{d\dot{\mathbf{y}}}{db} + \mathbf{Q}_z^A \frac{d\mathbf{z}}{db} - (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A) \right\} dt \\ &= - \left[\boldsymbol{\zeta}^T \mathbf{M} \frac{d\dot{\mathbf{z}}}{db} \right]_{t_0}^{t_f} + \int_{t_0}^{t_f} \left(\dot{\boldsymbol{\zeta}}^T \mathbf{M} + \boldsymbol{\zeta}^T \dot{\mathbf{M}} + \boldsymbol{\zeta}^T \mathbf{Q}_z^A \right) \frac{d\mathbf{z}}{db} dt \\ & \quad - \int_{t_0}^{t_f} \boldsymbol{\zeta}^T [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A] \frac{d\dot{\mathbf{y}}}{db} dt - \int_{t_0}^{t_f} \boldsymbol{\zeta}^T (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A) dt. \end{aligned} \quad (4.45)$$

Finally, take Equation (4.38) again multiplied by another set of adjoint variables $\boldsymbol{\xi}(t)$, without integration by parts:

$$\begin{aligned} & \int_{t_0}^{t_f} \boldsymbol{\xi}^T \left\{ -\mathbf{M} \frac{d\dot{\mathbf{z}}}{db} - [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A] \frac{d\dot{\mathbf{y}}}{db} + \mathbf{Q}_z^A \frac{d\mathbf{z}}{db} - (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A) \right\} dt \\ &= - \int_{t_0}^{t_f} \boldsymbol{\xi}^T \mathbf{M} \frac{d\dot{\mathbf{z}}}{db} dt - \int_{t_0}^{t_f} \boldsymbol{\xi}^T [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A] \frac{d\dot{\mathbf{y}}}{db} dt \\ & \quad + \int_{t_0}^{t_f} \boldsymbol{\xi}^T \mathbf{Q}_z^A \frac{d\mathbf{z}}{db} dt - \int_{t_0}^{t_f} \boldsymbol{\xi}^T (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A) dt. \end{aligned} \quad (4.46)$$

The aim of these operations becomes clear after adding (4.44) (4.45) and (4.46) to $\frac{d\bar{r}}{db}$ in Equation (4.43). After rearranging, adjoint variables can be selected that remove the terms with $\frac{dy}{db}$, $\frac{dz}{db}$ and $\frac{d\dot{z}}{db}$. Using symmetry of the mass matrix, the adjoint variables should satisfy:

$$\dot{\mu} = [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A]^T (\zeta + \xi) - p_y^T, \quad (4.47)$$

$$\mathbf{M}\dot{\zeta} = -\mu - \dot{\mathbf{M}}\zeta - \mathbf{Q}_z^A (\zeta + \xi) - p_z^T, \quad (4.48)$$

with boundary conditions:

$$\mu(t_f) = \mathbf{0}, \quad (4.49)$$

$$\zeta(t_f) = \mathbf{0}. \quad (4.50)$$

Herein, the 'intermediate' adjoint variables ξ follow from:

$$\mathbf{M}\xi = p_z^T. \quad (4.51)$$

The end conditions in Equations (4.49) and (4.50) require that the first-order differential equations (4.47) and (4.48) are integrated *backward* in time. Furthermore, Equation (4.43) becomes:

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} [p_b - (\zeta^T + \xi^T) (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A)] dt. \quad (4.52)$$

More adjoint equations follow if t_0 , t_f , $\frac{dy}{db}(t_0)$ and $\frac{dz}{db}(t_0)$ do not have a predefined constant value, but are explicitly or implicitly determined by the design variable values as well. Bestle and Eberhard (1992) elaborated the sensitivity analysis for ordinary differential equations of multibody systems in a quite general format.

According to Equation (4.42), a time point response yields Dirac delta functions in the right-hand sides of the adjoint equations (4.47), (4.48) and (4.51). Haftka and Gürdal (1992) remark that a simpler form can be derived without Dirac function, by integrating (4.47) and (4.48) from $t_p - \epsilon$ to $t_p + \epsilon$ for an infinitesimal ϵ . Hereto, adjoint variable ξ has to be equated from (4.51), and inserted in (4.47) and (4.48). Finally, the following adjoint equations are found:

$$\dot{\mu} = [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A]^T \zeta, \quad (4.53)$$

$$\mathbf{M}\dot{\zeta} = -\mu - \dot{\mathbf{M}}\zeta - \mathbf{Q}_z^A \zeta, \quad (4.54)$$

with end conditions:

$$\mu^T(t_p) = \{r_y - r_z \mathbf{M}^{-1} [(\mathbf{M}\dot{\mathbf{z}})_y - \mathbf{Q}_y^A]\}_{t=t_p}, \quad (4.55)$$

$$\zeta^T(t_p) = [(r_z + r_z \mathbf{M}^{-1} \mathbf{Q}_z^A) \mathbf{M}^{-1}]_{t=t_p}, \quad (4.56)$$

and the derivative of the response \bar{r} :

$$\frac{d\bar{r}}{db} = r_b(t_p) - \int_{t_0}^{t_p} \zeta^T (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A) dt - [r_{\dot{\mathbf{z}}} \mathbf{M}^{-1} (\mathbf{M}_b \dot{\mathbf{z}} - \mathbf{Q}_b^A)]_{t=t_p}. \quad (4.57)$$

4.4.3 Discussion

An easy way to calculate the multibody responses and design sensitivities is to use the direct method, and treat the first-order equations of motion and all sensitivity equations (4.35) to (4.38) as one large set of coupled ordinary differential equations. It requires a minimum of implementational efforts, and responses and derivatives will be calculated at the same time points, as required for time point constraints. However, the sensitivity equations (4.37)-(4.38) are decoupled for each design variable. This means that they can be integrated separately. In that case the time points for which sensitivities are calculated may not coincide, and an additional interpolation has to be applied.

The adjoint variable approach is often numerically more efficient if sensitivities are needed just for a small number of responses in combination with a large number of design variables. The adjoint equations follow independent of the number of design variables. They have to be numerically integrated backward in time. So, during forward integration of (4.35)-(4.36), one is *obliged* to store all relevant information that is necessary for backward integration of adjoint equations (4.47)-(4.48). Generally, time points of forward and backward integration do not coincide, and the saved information must be interpolated. This may be a source of error during the backward integration, which places high accuracy demands on the interpolation method. After the forward integration has been completed, the adjoint equations can be solved independently for each response.

4.5 Multibody Systems described by Differential-Algebraic Equations

4.5.1 Differential-Algebraic Equations of Motion

A general formulation of the multibody system equations of motion is:

$$\mathbf{M}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T(\mathbf{q}, \mathbf{b}, t) \boldsymbol{\lambda} = \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t), \quad (4.58)$$

$$\Phi(\mathbf{q}, \mathbf{b}, t) = \mathbf{0}. \quad (4.59)$$

Matrix \mathbf{M} and column \mathbf{Q}^A are the mass matrix and generalized applied force, respectively. Forces due to kinematic constraints (4.59) are expressed in the dynamic

equations of motion (4.58) by means of Lagrange multipliers λ . The generalized coordinates $\mathbf{q}(t)$ can be defined either by a Cartesian or a relative coordinate formulation. Velocity and acceleration equations must hold as well:

$$\Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}, t)\dot{\mathbf{q}} = \nu(\mathbf{q}, \mathbf{b}, t), \quad (4.60)$$

$$\Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}, t)\ddot{\mathbf{q}} = \gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t). \quad (4.61)$$

All together, the motion of the multibody system is represented by a mixed set of differential-algebraic equations.

The numerical solution of the differential-algebraic equations of motion is less straightforward compared with ordinary first-order differential equations. The difficulty of the numerical solution lies in the algebraic equations (4.59). Therefore, it is often tried to reduce the equations of motion towards ordinary differential equations. Then, an integration algorithm for first-order systems can be used, as outlined in Section 4.4. Within this context one may start from Equations (4.58) and (4.61):

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A \\ \gamma \end{bmatrix}. \quad (4.62)$$

Given an assembled configuration with initial conditions:

$$\mathbf{q}(t_0) = \mathbf{q}_0, \quad (4.63)$$

$$\dot{\mathbf{q}}(t_0) = \dot{\mathbf{q}}_0, \quad (4.64)$$

accelerations $\ddot{\mathbf{q}}(t_0)$ and Lagrange multipliers $\lambda(t_0)$ can be computed, provided that the mass matrix \mathbf{M} is positive definite and the Jacobian matrix $\Phi_{\mathbf{q}}$ has full row rank (Haug, 1989). Positions $\mathbf{q}(t_{p+1})$ and velocities $\dot{\mathbf{q}}(t_{p+1})$ at a next time instant t_{p+1} ($p = 0$ at the first time step) follow from numerical integration of the accelerations. Afterwards, Equation (4.62) can be solved again to obtain $\ddot{\mathbf{q}}(t_{p+1})$ and $\lambda(t_{p+1})$. This process is continued until t_{p+1} exceeds the final time.

The abovementioned direct integration procedure can be unstable. Large errors in (4.59) and (4.60) may arise. Therefore, an additional constraint stabilization or correction is necessary. A well-known approach in multibody system dynamics is generalized coordinate partitioning. It determines dependent and independent generalized coordinates from the Jacobian matrix. The independent accelerations are integrated, after which the dependent variables are evaluated by solving kinematic position and velocity equations (4.59) and (4.60). Generalized coordinate partitioning has proven to be reliable and accurate, although it is numerically expensive due to the iterative solution of the position equations at each time step. A detailed description is given by Haug (1989). Three references in Table 4.1 use this method: Haug *et al.* (1984), Haug (1987), and Ashrafioun and Mani (1990).

An other constraint stabilization technique is Baumgarte's method. This technique is applied in Chang and Nikravesh (1985), Dias and Pereira (1994), and Pagalday *et al.* (1996). Right-hand side γ in Equation (4.62) is replaced by:

$$\hat{\gamma} = \gamma - 2\alpha(\Phi_{\mathbf{q}}\dot{\mathbf{q}} - \nu) - \beta^2\Phi, \quad (4.65)$$

in which the position and velocity equations are accounted for. The modified acceleration equation is stable for $\alpha > 0$ and $\beta \neq 0$. In first-order form direct integration can be carried out using algorithms of Section 4.4. However, the appropriate values of α and β are not known beforehand, and divergence may occur near singular configurations (Haug, 1989). Therefore, in DADS (CADSi, 1995) a hybrid algorithm is implemented combining the efficiency of Baumgarte's method and the reliability of generalized coordinate partitioning.

The multibody package MECANO (Samtech, 1994) uses a completely different approach to solve the differential-algebraic equations of motion. An implicit Newmark algorithm is employed. For each new time point t_p during integration, Equations (4.58) and (4.59) are iteratively solved, using the formulas of Newmark to approximate $\dot{\mathbf{q}}(t_{p+1})$ and $\ddot{\mathbf{q}}(t_{p+1})$ in terms of $\mathbf{q}(t_p)$, $\dot{\mathbf{q}}(t_p)$, $\ddot{\mathbf{q}}(t_p)$ and $\mathbf{q}(t_{p+1})$. A Newton-Raphson procedure is used to calculate $\mathbf{q}(t_{p+1})$ and $\lambda(t_{p+1})$. No further stabilization from (4.60) or (4.61) is required. The Newmark algorithm is explained in detail in e.g. G eradin and Rixen (1994).

4.5.2 Second-Order Differential-Algebraic Sensitivity Equations

Derivation of the sensitivity equations proceeds in a similar manner as shown in Section 4.4.2. The equations only become slightly more complex, especially the adjoint sensitivity equations. Most references maintain the second-order form of the differential-algebraic equations of motion. Direct differentiation of Equations (4.58) and (4.59) gives:

$$\mathbf{M}\frac{d\ddot{\mathbf{q}}}{db} + \Phi_{\mathbf{q}}^T\frac{d\lambda}{db} = -[(\mathbf{M}\ddot{\mathbf{q}})_{\mathbf{q}} + (\Phi_{\mathbf{q}}^T\lambda)_{\mathbf{q}} - \mathbf{Q}_{\mathbf{q}}^A]\frac{d\mathbf{q}}{db} + \mathbf{Q}_{\dot{\mathbf{q}}}^A\frac{d\dot{\mathbf{q}}}{db} - (\mathbf{M}_b\ddot{\mathbf{q}} + \Phi_{\mathbf{q}b}^T\lambda - \mathbf{Q}_b^A), \quad (4.66)$$

$$\Phi_{\mathbf{q}}\frac{d\mathbf{q}}{db} = -\Phi_b, \quad (4.67)$$

with initial conditions:

$$\frac{d\mathbf{q}}{db}(t_0) = \mathbf{0}, \quad (4.68)$$

$$\frac{d\dot{\mathbf{q}}}{db}(t_0) = \mathbf{0}. \quad (4.69)$$

According to Equations (4.24) and (4.25), the following velocity and acceleration sensitivity equations hold as well:

$$\Phi_{\mathbf{q}} \frac{d\dot{\mathbf{q}}}{db} = - [(\Phi_{\mathbf{q}\dot{\mathbf{q}}})_{\mathbf{q}} - \nu_{\mathbf{q}}] \frac{d\mathbf{q}}{db} - (\Phi_{\mathbf{q}b\dot{\mathbf{q}}} - \nu_b), \quad (4.70)$$

$$\Phi_{\mathbf{q}} \frac{d\ddot{\mathbf{q}}}{db} = - [(\Phi_{\mathbf{q}\ddot{\mathbf{q}}})_{\mathbf{q}} - \gamma_{\mathbf{q}}] \frac{d\mathbf{q}}{db} + \gamma_{\dot{\mathbf{q}}} \frac{d\dot{\mathbf{q}}}{db} - (\Phi_{\mathbf{q}b\ddot{\mathbf{q}}} - \gamma_b). \quad (4.71)$$

Hence, a set of linear second-order differential-algebraic equations determines the position, velocity, acceleration and Lagrange multiplier design sensitivities. They can be solved for each design variable $b \in \mathbf{b}$, using the previously mentioned solution procedures. This direct differentiation method is, for example, applied in Ashrafiuon and Mani (1990), and Dias and Pereira (1994).

To derive the adjoint equations, the procedure outlined in Haug (1987) is followed, although design dependent initial conditions and final time are not included here. Fleury (1996) also adopts an approach similar to Haug (1987). We take as starting point the response:

$$\bar{r} = \int_{t_0}^{t_f} p(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \mathbf{b}, t) dt, \quad (4.72)$$

The corresponding design sensitivity is:

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} \left(p_{\mathbf{q}} \frac{d\mathbf{q}}{db} + p_{\dot{\mathbf{q}}} \frac{d\dot{\mathbf{q}}}{db} + p_{\ddot{\mathbf{q}}} \frac{d\ddot{\mathbf{q}}}{db} + p_{\boldsymbol{\lambda}} \frac{d\boldsymbol{\lambda}}{db} + p_b \right) dt, \quad (4.73)$$

which becomes after two times integration by parts:

$$\begin{aligned} \frac{d\bar{r}}{db} = & \int_{t_0}^{t_f} \left(p_{\mathbf{q}} - \frac{dp_{\dot{\mathbf{q}}}}{dt} + \frac{d^2 p_{\ddot{\mathbf{q}}}}{dt^2} \right) \frac{d\mathbf{q}}{db} dt + \int_{t_0}^{t_f} p_{\boldsymbol{\lambda}} \frac{d\boldsymbol{\lambda}}{db} dt \\ & + \int_{t_0}^{t_f} p_b dt + \left[\left(p_{\dot{\mathbf{q}}} - \frac{dp_{\ddot{\mathbf{q}}}}{dt} \right) \frac{d\mathbf{q}}{db} \right]_{t_0}^{t_f} + \left[p_{\ddot{\mathbf{q}}} \frac{d\dot{\mathbf{q}}}{db} \right]_{t_0}^{t_f}. \end{aligned} \quad (4.74)$$

Now equations have to be added that avoid the evaluation of $\frac{d\mathbf{q}}{db}$, $\frac{d\boldsymbol{\lambda}}{db}$, $\frac{d\mathbf{q}}{db}(t_f)$ and $\frac{d\dot{\mathbf{q}}}{db}(t_f)$ in (4.74). For this purpose, the following recipe is applied. Multiply Equation (4.66) by adjoint variables $\boldsymbol{\mu}(t)$, and two times integrate by parts. Furthermore, multiply Equation (4.67) by adjoint variables $\boldsymbol{\eta}(t)$, and integrate from t_0 to t_f (no integration by parts). Two more equations are necessary: the position and velocity sensitivity equations (4.67) and (4.70) at final time t_f , multiplied by adjoint variables $\boldsymbol{\tau}^f$ and $\boldsymbol{\kappa}^f$, respectively. Finally, add all expressions to Equation (4.74), rearrange, and select the adjoint variables.

Assuming symmetry of the mass matrix, the following adjoint relations are obtained:

$$\begin{aligned} \mathbf{M}\ddot{\boldsymbol{\mu}} + \left(2\dot{\mathbf{M}} + \mathbf{Q}_q^A\right)^T \dot{\boldsymbol{\mu}} + \left\{ \dot{\mathbf{M}} + \frac{d\mathbf{Q}_q^A}{dt} + [(\mathbf{M}\ddot{\mathbf{q}})_q + (\Phi_q^T \boldsymbol{\lambda})_q - \mathbf{Q}_q^A] \right\}^T \boldsymbol{\mu} \\ + \Phi_q^T \boldsymbol{\eta} = p_q^T - \frac{dp_q^T}{dt} + \frac{d^2 p_q^T}{dt^2}, \end{aligned} \quad (4.75)$$

$$\Phi_q \boldsymbol{\mu} = p_\lambda^T, \quad (4.76)$$

and:

$$\begin{aligned} \mathbf{M}(t_f) \dot{\boldsymbol{\mu}}(t_f) + \left[\dot{\mathbf{M}}(t_f) + \mathbf{Q}_q^A(t_f) \right]^T \boldsymbol{\mu}(t_f) - [(\Phi_q \dot{\mathbf{q}})_q - \boldsymbol{\nu}_q]_{t=t_f}^T \boldsymbol{\kappa}^f \\ - \Phi_q^T(t_f) \boldsymbol{\tau}^f = -p_q^T(t_f) + \frac{dp_q^T}{dt}(t_f), \end{aligned} \quad (4.77)$$

$$\mathbf{M}(t_f) \boldsymbol{\mu}(t_f) + \Phi_q^T(t_f) \boldsymbol{\kappa}^f = p_q^T(t_f), \quad (4.78)$$

with:

$$\begin{aligned} \frac{d\bar{r}}{db} = \int_{t_0}^{t_f} \left[p_b - \boldsymbol{\mu}^T (\mathbf{M}_b \ddot{\mathbf{q}} + \Phi_{qb}^T \boldsymbol{\lambda} - \mathbf{Q}_b^A) - \boldsymbol{\eta}^T \Phi_b \right] dt \\ - \boldsymbol{\tau}^{fT} \Phi_b(t_f) - \boldsymbol{\kappa}^{fT} [(\Phi_{qb}(t_f) \dot{\mathbf{q}}(t_f) - \boldsymbol{\nu}_b(t_f))]. \end{aligned} \quad (4.79)$$

The numerical solution comprises of four stages. Firstly, integrate the differential-algebraic equations of motion (4.58) to (4.61) forward in time, calculate \bar{r} from (4.72), and store all data necessary for the adjoint equations. Next, solve $\boldsymbol{\mu}(t_f) = \boldsymbol{\mu}^f$ and $\boldsymbol{\kappa}^f$ from Equations (4.78) and (4.76):

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix}_{t=t_f} \begin{bmatrix} \boldsymbol{\mu}^f \\ \boldsymbol{\kappa}^f \end{bmatrix} = \begin{bmatrix} p_q^T \\ p_\lambda^T \end{bmatrix}_{t=t_f}. \quad (4.80)$$

Afterwards, solve $\dot{\boldsymbol{\mu}}(t_f) = \dot{\boldsymbol{\mu}}^f$ and $\boldsymbol{\tau}^f$ from Equation (4.77) and the time derivative of Equation (4.76):

$$\begin{aligned} \begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix}_{t=t_f} \begin{bmatrix} \dot{\boldsymbol{\mu}}^f \\ -\boldsymbol{\tau}^f \end{bmatrix} = \begin{bmatrix} -(\dot{\mathbf{M}} + \mathbf{Q}_q^A)^T & [(\Phi_q \dot{\mathbf{q}})_q - \boldsymbol{\nu}_q]^T \\ -\frac{d\Phi_q}{dt} & \mathbf{0} \end{bmatrix}_{t=t_f} \begin{bmatrix} \boldsymbol{\mu}^f \\ \boldsymbol{\kappa}^f \end{bmatrix} \\ + \begin{bmatrix} -p_q^T + \frac{dp_q^T}{dt} \\ \frac{dp_\lambda^T}{dt} \end{bmatrix}_{t=t_f}. \end{aligned} \quad (4.81)$$

Finally, integrate the linear second-order differential-algebraic equations (4.75) and (4.76) backward in time with end conditions:

$$\bar{\boldsymbol{\mu}}(t_f) = \boldsymbol{\mu}^f, \quad (4.82)$$

$$\dot{\bar{\boldsymbol{\mu}}}(t_f) = \dot{\boldsymbol{\mu}}^f, \quad (4.83)$$

and calculate the derivative $\frac{d\bar{r}}{db}$ from (4.79). The same solver may be used as was applied for forward integration of the equations of motion.

4.5.3 First-Order Ordinary Differential Sensitivity Equations

In Bestle and Seybold (1992) a different stand of view is taken. The equations of motion are formulated in first-order form. Furthermore, the acceleration equations (4.61) are used instead of the position equations (4.59) to derive the sensitivity equations:

$$\frac{d\dot{\mathbf{y}}}{db} = \frac{d\mathbf{z}}{db}, \quad (4.84)$$

$$\mathbf{M} \frac{d\dot{\mathbf{z}}}{db} + \Phi_{\mathbf{y}}^T \frac{d\boldsymbol{\lambda}}{db} = - [(\mathbf{M}\dot{\mathbf{z}})_{\mathbf{y}} + (\Phi_{\mathbf{y}}^T \boldsymbol{\lambda})_{\mathbf{y}} - \mathbf{Q}_{\mathbf{y}}^A] \frac{d\mathbf{y}}{db} + \mathbf{Q}_{\mathbf{z}}^A \frac{d\mathbf{z}}{db} - (\mathbf{M}_b \dot{\mathbf{z}} + \Phi_{\mathbf{y}b}^T \boldsymbol{\lambda} - \mathbf{Q}_b^A), \quad (4.85)$$

$$\Phi_{\mathbf{y}} \frac{d\dot{\mathbf{z}}}{db} = - [(\Phi_{\mathbf{y}} \dot{\mathbf{z}})_{\mathbf{y}} - \gamma_{\mathbf{y}}] \frac{d\mathbf{y}}{db} + \gamma_{\mathbf{z}} \frac{d\mathbf{z}}{db} - (\Phi_{\mathbf{y}b} \dot{\mathbf{z}} - \gamma_b), \quad (4.86)$$

with $\mathbf{y} = \mathbf{q}$ and $\mathbf{z} = \dot{\mathbf{q}}$, and initial conditions:

$$\frac{d\mathbf{y}}{db}(t_0) = \mathbf{0}, \quad (4.87)$$

$$\frac{d\mathbf{z}}{db}(t_0) = \mathbf{0}. \quad (4.88)$$

Bestle and Seybold (1992) assumed that the numerical solution of the sensitivity equations do not need any further stabilization.

The corresponding adjoint equations are first-order ordinary differential equations. We start again from:

$$\bar{r} = \int_{t_0}^{t_f} p(\mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \boldsymbol{\lambda}, \mathbf{b}, t) dt, \quad (4.89)$$

and differentiate with respect to b :

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} \left(p_{\mathbf{y}} \frac{d\mathbf{y}}{db} + p_{\mathbf{z}} \frac{d\mathbf{z}}{db} + p_{\dot{\mathbf{z}}} \frac{d\dot{\mathbf{z}}}{db} + p_{\boldsymbol{\lambda}} \frac{d\boldsymbol{\lambda}}{db} + p_b \right) dt. \quad (4.90)$$

Add Equations (4.84), (4.85), and (4.86), multiplied by adjoint variables $\mu(t)$, $\xi(t)$, and $\eta(t)$, respectively. Furthermore, add Equation (4.85) multiplied by $\zeta(t)$ and integrate by parts. Then, assuming that $\mathbf{M} = \mathbf{M}^T$, the following adjoint relations are obtained:

$$\dot{\mu} = [(\mathbf{M}\dot{\mathbf{z}})_y + (\Phi_y^T \lambda)_y - \mathbf{Q}_y^A]^T (\zeta + \xi) + [(\Phi_y \dot{\mathbf{z}})_y - \gamma_y]^T \eta - p_y^T, \quad (4.91)$$

$$\mathbf{M}\dot{\zeta} = -\mu - \dot{\mathbf{M}}\zeta - \mathbf{Q}_z^A (\zeta + \xi) - \mathbf{y}_z^T \eta - p_z^T, \quad (4.92)$$

with end conditions:

$$\mu(t_f) = \mathbf{0}, \quad (4.93)$$

$$\zeta(t_f) = \mathbf{0}. \quad (4.94)$$

Herein, the 'intermediate' adjoint variables ξ and η are determined by:

$$\begin{bmatrix} \mathbf{M} & \Phi_y^T \\ \Phi_y & \mathbf{0} \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} p_z^T \\ p_\lambda^T - \Phi_y \zeta \end{bmatrix}. \quad (4.95)$$

Equation (4.90) becomes:

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} \left[p_b - (\zeta^T + \xi^T) (\mathbf{M}_b \dot{\mathbf{z}} + \Phi_{yb}^T \lambda - \mathbf{Q}_b^A) - \eta^T (\Phi_{yb} \dot{\mathbf{z}} - \gamma_b) \right] dt. \quad (4.96)$$

Remind that there is no guarantee to have a stable numerical integration. Bestle and Seybold (1992) reported, however, that they studied several applications, and that they did not find any stability problems. Therefore, they concluded that the proposed method is stable.

4.5.4 Discussion

The points made in the discussion of Section 4.4.3 are valid here as well. Either the direct or the adjoint method can be applied to derive the sensitivity equations. The direct method is straightforward and simple, but can become computationally inefficient for growing number of design variables. Then, the somewhat more intricate and subtle adjoint method may be beneficial. However, the adjoint method suffers from three drawbacks. Firstly, it is far more difficult to implement than the direct approach. Secondly, a lot of input/output operations and data storage is required. And thirdly, it is rather sensitive for interpolation errors. On account of the latter observation, Bestle (1994) suggested to do the forward integration of the equations

of motion two to ten times as accurate as the backward integration of the adjoint sensitivity equations.

Two different sets of adjoint equations have been presented. The approach of Haug (1987) yields differential-algebraic adjoint equations, whereas the method of Bestle and Seybold (1992) gives pure ordinary differential equations. The latter set of equations is of course preferable, since they are easier to solve. Though, some caution is called for, since numerical stability of these adjoint equations has not been proven.

Another distinction is the way Haug (1987) on the one hand, and Bestle and Seybold (1992) on the other, avoid to explicitly calculate the position, velocity, acceleration and Lagrange multiplier design sensitivities. Haug (1987) starts with two times integration by parts of $\frac{d\mathbf{r}}{db}$, whereas Bestle and Seybold (1992) keep the original form of Equation (4.90). As a consequence, other equations have to be selected to remove the previously mentioned design sensitivities, and the resulting adjoint equations are different. The procedure of Haug (1987), for example, yields time derivatives of $p_{\mathbf{q}}$, $p_{\dot{\mathbf{q}}}$ and p_{λ} , which may be problematic in the case of time point responses. On the contrary, Dirac functions in the right-hand sides of the adjoint equations of Bestle and Seybold (1992) can be eliminated analogously to Section 4.4.2.

4.6 Aspects of Practical Implementation

Whatever the type of the equations of motion, the sensitivity equations require a lot of new terms to be developed. Additional to the terms present in the kinematic and dynamic equations:

$$\Phi_{\mathbf{q}}, \quad \nu, \quad \gamma, \quad \mathbf{M}, \quad \mathbf{Q}^A, \quad (4.97)$$

the following partial derivatives with respect to \mathbf{q} and $\dot{\mathbf{q}}$ need to be computed:

$$\nu_{\mathbf{q}}, \quad (\Phi_{\mathbf{q}}\ddot{\mathbf{q}})_{\mathbf{q}}, \quad \gamma_{\mathbf{q}}, \quad (\mathbf{M}\ddot{\mathbf{q}})_{\mathbf{q}}, \quad (\Phi_{\mathbf{q}}^T\lambda)_{\mathbf{q}}, \quad \mathbf{Q}_{\mathbf{q}}^A, \quad \gamma_{\dot{\mathbf{q}}}, \quad \mathbf{Q}_{\dot{\mathbf{q}}}^A. \quad (4.98)$$

Furthermore, partial derivatives with respect to the design variables arise:

$$\Phi_b, \quad \Phi_{q_b}, \quad \nu_b, \quad \gamma_b, \quad \mathbf{M}_b, \quad \mathbf{Q}_b^A. \quad (4.99)$$

In a general purpose program, the number of new terms grows very rapidly due to the large amount of design variables that can be identified for various applications. That is why many researchers apply symbolic computation to generate the sensitivity equations, instead of extending a library of standard joint elements.

However, many existing multibody analysis codes do not use symbolic computation, and numerically evaluate the equations of motion from a library of joint elements. Then, two alternatives remain for the generation of the sensitivity equations. Firstly, one can refrain from analytically deriving the terms with partial derivatives, but use finite differencing to calculate them. This is called the semi-analytical method, which is commonly applied in structural optimization to compute the derivatives of the stiffness and mass matrix with respect to the design variables. Of course, semi-analytically obtained design sensitivities will be less accurate than their analytical counterparts. It has to be investigated whether for multibody systems in general an acceptable accuracy remains. The other alternative is to apply automatic differentiation (recall the introduction of this chapter). Usually, each of the elements in the joint library is represented by a piece of Fortran or C code of limited length. Automatic differentiation may be able to produce new code for the extra partial derivatives in the sensitivity equations.

Another aspect of implementation concerns the calculation of integral functions \bar{r} and $\frac{d\bar{r}}{db}$ in Sections 4.4 and 4.5. They may be evaluated after the numerical solution of the differential(-algebraic) equations of motion and sensitivity equations has been completed. Bestle and Eberhard (1992) remarked, however, that it is computationally convenient to rewrite \bar{r} and $\frac{d\bar{r}}{db}$ into differential equations as well. Suppose that we have:

$$\bar{u} = \int_{t_0}^{t_f} v dt, \quad (4.100)$$

and that function v depends on the solution of the forward integration of some set of differential equations. Then, Equation (4.100) can be rewritten as an initial value problem:

$$\begin{aligned} \dot{u} &= v, & u(t_0) &= 0, \\ \bar{u} &= u(t_f), \end{aligned} \quad (4.101)$$

and solved along with the other differential equations. In the case of backward integration one can use:

$$\begin{aligned} \dot{u} &= -v, & u(t_f) &= 0, \\ \bar{u} &= u(t_0). \end{aligned} \quad (4.102)$$

The initial value problem formulation of Equation (4.101) is useful for the direct method. The final value problem of Equation (4.102) is suited for the adjoint method.

4.7 Examples

4.7.1 Impact-Absorber

Consider again the impact-absorber of Section 2.5.1. The equations of motion are in first-order form:

$$\dot{y} = z, \quad y(0) = 0, \quad (4.103)$$

$$\dot{z} = -b_1 y - b_2 z, \quad z(0) = 1. \quad (4.104)$$

The variables y and z represent the mass position and velocity. The acceleration response \dot{z} at an arbitrarily chosen time point $t_p = 5$ s and comfort criterion (2.34) are of main interest. The objective is to calculate the corresponding design sensitivities for a stiffness coefficient of $b_1 = 0.5 \text{ Nm}^{-1}$ and a damping coefficient of $b_2 = 0.5 \text{ Nsm}^{-1}$. For this purpose, three different approaches are applied: finite differencing, the direct method, and the adjoint method. Each approach requires first-order ordinary differential equations to be solved, either forward or backward in time. To this end, a fourth-order Runge-Kutta algorithm of MATLAB (MathWorks, 1996) is applied at a tolerance level of 10^{-6} . Furthermore, cubic spline interpolation is used if a state variable value is needed at a time point that does not coincide with the mesh points of integration.

Time Point Acceleration

Firstly, derivatives of the acceleration with respect to the design variables are calculated at time point $t_p = 5$ s. Finite difference sensitivities are obtained by applying forward difference formula (4.1). This means that the equations of motion are solved for the original design as well as for two perturbed designs. Then, acceleration values of six digits are calculated by interpolation. A range of perturbations Δb is applied, and the finite difference sensitivities are compared with the analytical solution. The relative error is plotted in Figure 4.1 for both design variables. The smallest errors are found near $\Delta b = 10^{-3}$, which corresponds with the optimal step size resulting from Equation (4.5). For small perturbations the condition error dominates, for large perturbations the truncation error.

Direct differentiation yields the following sensitivity equations for b_1 :

$$\frac{d\dot{y}}{db_1} = \frac{dz}{db_1}, \quad \frac{dy}{db_1}(t_0) = 0, \quad (4.105)$$

$$\frac{d\dot{z}}{db_1} = -b_1 \frac{dy}{db_1} - b_2 \frac{dz}{db_1} - y, \quad \frac{dz}{db_1}(t_0) = 0, \quad (4.106)$$

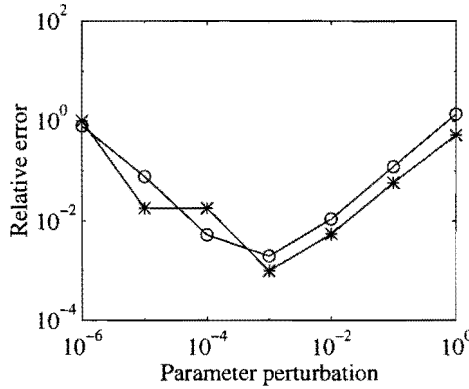


Figure 4.1: Relative errors of the acceleration sensitivities at $t_p = 5$ s computed by forward finite difference as a function of the parameter perturbations. Errors of the derivatives with respect to b_1 and b_2 correspond with \circ and $*$, respectively.

and for b_2 :

$$\frac{dy}{db_2} = \frac{dz}{db_2}, \quad \frac{dy}{db_2}(t_0) = 0, \quad (4.107)$$

$$\frac{dz}{db_2} = -b_1 \frac{dy}{db_2} - b_2 \frac{dz}{db_2} - z, \quad \frac{dz}{db_2}(t_0) = 0. \quad (4.108)$$

They are solved together with the equations of motion as one set of ordinary differential equations. The relative accuracy of the interpolated derivatives $\frac{dz}{db_1}(t_p)$ and $\frac{dz}{db_2}(t_p)$ are given in Table 4.2. The accuracy is much higher than can be obtained by finite differencing.

Acceleration sensitivity	Analytical solution	Relative error	
		Direct	Adjoint
$\frac{dz}{db_1}(t_p)$ [N ⁻¹ s ⁻² m ²]	0.55708928	$1.41 \cdot 10^{-6}$	$2.22 \cdot 10^{-6}$
$\frac{dz}{db_2}(t_p)$ [N ⁻¹ s ⁻³ m ²]	-0.29470796	$1.48 \cdot 10^{-6}$	$2.34 \cdot 10^{-6}$

Table 4.2: Relative errors of the acceleration sensitivities at $t_p = 5$ s computed by the direct and the adjoint method.

The adjoint method uses $p = \dot{z}\delta(t - t_p)$ in Equation (4.42). Then, Equations (4.53) to (4.56) yield the adjoint equations:

$$\dot{\mu} = b_1 \zeta, \quad \mu(t_p) = -b_1, \quad (4.109)$$

$$\dot{\zeta} = -\mu + b_2 \zeta, \quad \zeta(t_p) = -b_2, \quad (4.110)$$

with the design sensitivities defined by Equation (4.57):

$$\frac{d\dot{z}}{db_1}(t_p) = - \int_{t_0}^{t_p} y\zeta dt - y(t_p), \quad (4.111)$$

$$\frac{d\dot{z}}{db_2}(t_p) = - \int_{t_0}^{t_p} z\zeta dt - z(t_p). \quad (4.112)$$

Using Equation (4.102), these integral functions are rewritten as:

$$\frac{d\dot{u}}{db_1} = y\zeta, \quad \frac{du}{db_1}(t_p) = -y(t_p), \quad (4.113)$$

$$\frac{d\dot{u}}{db_2} = z\zeta, \quad \frac{du}{db_2}(t_p) = -z(t_p), \quad (4.114)$$

such that they can be solved backward in time along with the adjoint equations. Afterwards, the derivatives of the acceleration follow from:

$$\frac{d\dot{z}}{db_1}(t_p) = \frac{du}{db_1}(t_0), \quad (4.115)$$

$$\frac{d\dot{z}}{db_2}(t_p) = \frac{du}{db_2}(t_0). \quad (4.116)$$

About the same accuracy is found as with the direct approach. For this example the adjoint method is computationally more expensive than the direct method due to the interpolation during backward integration. The adjoint approach will be most efficient for a larger number of design variables.

Comfort Criterion

Similarly, the design sensitivities of the comfort criterion:

$$F = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \dot{z}^2 dt \quad (4.117)$$

are computed. Forward finite differencing gives Figure 4.2. The optimum step size lies again near $\Delta b = 10^{-3}$, however the relative error of $\frac{dF}{db_1}$ is smaller than the error of $\frac{dF}{db_2}$. The explanation can be found in Figure 2.5: the comfort criterion behaves more linearly in b_1 -direction (stiffness) than in b_2 -direction (damping).

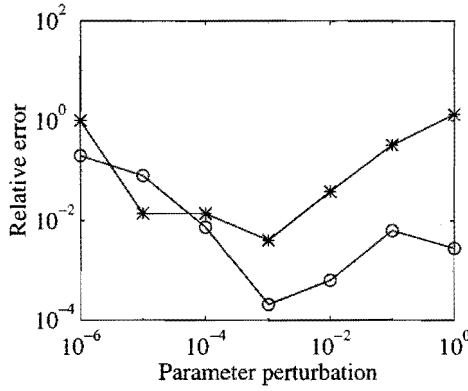


Figure 4.2: Relative errors of the comfort criterion sensitivities computed by forward finite difference as a function of the parameter perturbations. Errors of the derivatives with respect to b_1 and b_2 correspond with \circ and $*$, respectively.

Using the direct differentiation approach, Equations (4.103) to (4.108) are integrated forward in time together with:

$$\dot{u} = \frac{\dot{z}^2}{t_f - t_0}, \quad u(t_0) = 0, \quad (4.118)$$

$$\frac{d\dot{u}}{db_1} = \frac{2\dot{z}}{t_f - t_0} \frac{d\dot{z}}{db_1}, \quad \frac{du}{db_1}(t_0) = 0, \quad (4.119)$$

$$\frac{d\dot{u}}{db_2} = \frac{2\dot{z}}{t_f - t_0} \frac{d\dot{z}}{db_2}, \quad \frac{du}{db_2}(t_0) = 0. \quad (4.120)$$

The comfort value and the design sensitivities then follow from:

$$F = u(t_f), \quad (4.121)$$

$$\frac{dF}{db_1} = \frac{du}{db_1}(t_f), \quad (4.122)$$

$$\frac{dF}{db_2} = \frac{du}{db_2}(t_f). \quad (4.123)$$

In Table 4.3 the relative errors of the computed sensitivities are presented.

The adjoint method first integrates the equations of motion (4.103) and (4.104) forward in time completed with:

$$\dot{u} = \frac{\dot{z}^2}{t_f - t_0}, \quad u(t_0) = 0. \quad (4.124)$$

Comfort sensitivity	Analytical solution	Relative error	
		Direct	Adjoint
$\frac{dF}{db_1}$ [$N^{-1}s^{-4}m^3$]	0.083382659	$1.05 \cdot 10^{-6}$	$2.80 \cdot 10^{-5}$
$\frac{dF}{db_2}$ [$N^{-1}s^{-5}m^3$]	-0.040564101	$1.26 \cdot 10^{-6}$	$4.19 \cdot 10^{-5}$

Table 4.3: Relative errors of the comfort criterion sensitivities computed by the direct and the adjoint method.

The comfort criterion then is:

$$F = u(t_f). \quad (4.125)$$

Afterwards the adjoint equations have to be integrated backward in time:

$$\dot{\mu} = b_1(\zeta + \xi), \quad \mu(t_f) = 0, \quad (4.126)$$

$$\dot{\zeta} = -\mu + b_2(\zeta + \xi), \quad \zeta(t_f) = 0, \quad (4.127)$$

$$\frac{d\dot{u}}{db_1} = y(\zeta + \xi), \quad \frac{du}{db_1}(t_f) = 0, \quad (4.128)$$

$$\frac{d\dot{u}}{db_2} = z(\zeta + \xi), \quad \frac{du}{db_2}(t_f) = 0. \quad (4.129)$$

Herein, intermediate variable ξ is equal to:

$$\xi = \frac{2\dot{z}}{t_f - t_0}. \quad (4.130)$$

Finally, the comfort criterion sensitivities result:

$$\frac{dF}{db_1} = \frac{du}{db_1}(t_0), \quad (4.131)$$

$$\frac{dF}{db_2} = \frac{du}{db_2}(t_0). \quad (4.132)$$

Now, the computed sensitivities are more than ten times less accurate than the sensitivities of direct differentiation. This is mainly due to the interpolation of the state variables y and z during backward integration of the adjoint equations.

4.7.2 Slider-Crank Mechanism

Design sensitivity analysis of kinematically driven multibody systems is demonstrated using the slider-crank mechanism of Section 2.5.2. Exactly the same optimization problem is considered, but now the position equations of motion are

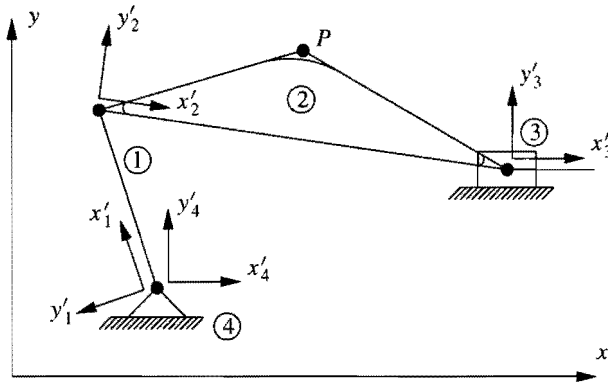


Figure 4.3: Definition of the global reference frame $x-y$ and body-fixed reference frames $x'_i-y'_i$ in the slider-crank mechanism. Definition of the design variables can be found in Figure 2.6.

assembled starting from a Cartesian coordinate formulation. Four bodies are identified: 1) the input-crank, 2) the coupler, 3) the slider, and 4) the ground. Each body has its own body-fixed reference frame $x'_i-y'_i$ as indicated in Figure 4.3.

Eleven algebraic equations arise from the restrictions with respect to the body movements. The ground is fixed to the global $x-y$ frame, yielding:

$$\Phi_1 = x_4 - b_5 = 0, \quad (4.133)$$

$$\Phi_2 = y_4 - b_6 = 0, \quad (4.134)$$

$$\Phi_3 = \theta_4 = 0. \quad (4.135)$$

Furthermore, revolute joints are present between the ground and the input crank:

$$\Phi_4 = x_1 - x_4 = 0, \quad (4.136)$$

$$\Phi_5 = y_1 - y_4 = 0, \quad (4.137)$$

between the input crank and the coupler:

$$\Phi_6 = x_1 + b_1 \cos \theta_1 - x_2 = 0, \quad (4.138)$$

$$\Phi_7 = y_1 + b_1 \sin \theta_1 - y_2 = 0, \quad (4.139)$$

and between the coupler and the slider:

$$\Phi_8 = x_2 + b_2 \cos \theta_2 - x_3 = 0, \quad (4.140)$$

$$\Phi_9 = y_2 + b_2 \sin \theta_2 - y_3 = 0. \quad (4.141)$$

The translational joint between the slider and the ground is defined by:

$$\Phi_{10} = y_3 - b_3 = 0, \quad (4.142)$$

$$\Phi_{11} = \theta_3 = 0. \quad (4.143)$$

Since there are twelve generalized coordinates, one degree of freedom remains.

To be able to calculate the generalized coordinates:

$$\mathbf{q}^T = [x_1 \quad y_1 \quad \theta_1 \quad x_2 \quad \dots \quad \theta_4] \quad (4.144)$$

for a crank rotation $\Delta\gamma$, one additional driving constraint is specified:

$$\Phi_{12} = \theta_1 - (b_8 + \Delta\gamma) = 0. \quad (4.145)$$

The generalized coordinates are solved using a Newton-Raphson iteration process. Starting from an initial guess \mathbf{q}^0 , the estimate \mathbf{q}^l of the l -th iteration is improved by:

$$\Phi_{\mathbf{q}}(\mathbf{q}^l, \Delta\gamma)\Delta\mathbf{q}^l = -\Phi(\mathbf{q}^l, \Delta\gamma), \quad (4.146)$$

$$\mathbf{q}^{l+1} = \mathbf{q}^l + \Delta\mathbf{q}^l. \quad (4.147)$$

This process is continued until the change of generalized coordinates and the violation of the algebraic kinematic relations is smaller than 10^{-6} .

The sensitivity analysis has to provide the derivatives of θ_2 with respect to the design variables for all crank rotations $\Delta\gamma$ given in Table 2.2. The direct method uses Equation (4.15). Since there are eight crank rotations, and six design variables present in the algebraic equations, a total amount of 48 matrix equations has to be solved. Here, it is efficient to re-use the decomposition of the Jacobian matrix $\Phi_{\mathbf{q}}$. On the contrary, the adjoint method starts from Equation (4.21) with \mathbf{w} a column of zeros, except for the sixth element that is equal to 1. Now, only 8 linear matrix equations need to be solved, followed by 48 column multiplications according to Equation (4.22) to finally obtain the required sensitivities.

A position and sensitivity analysis is carried out with the following initial guess for the first crank rotation $\Delta\gamma = 0$:

$$\mathbf{q}^{0T} = [3 \quad 4 \quad 2 \quad -2 \quad 2 \quad -1 \quad 20 \quad -4 \quad 0 \quad 3 \quad 4 \quad 0]. \quad (4.148)$$

For each subsequent crank rotation the new initial guess is equal to the generalized coordinates computed for the previous rotation. Then, the computational cost to calculate the sensitivities of θ_2 for all crank rotations is less than half the cost of the

complete position analysis if the direct method is applied. The cost of the adjoint method is even less than a tenth. Compare this with an overall finite difference method, which would require a multitude of the computational cost of the position analysis.

For this example all design variables are linearly present in the algebraic constraint equations. Therefore, Φ_b can be calculated semi-analytically without difficulty provided that the finite difference step is not too small. If the angle of the sliding axis is defined as a design variable, then Equation (4.142) is replaced by an algebraic relation that includes cosine and sine terms of this variable, (see Haug, 1989). In that case the magnitude of the difference step should be selected more carefully for this variable.

4.8 Conclusion and Discussion

Three different approaches have been presented to compute the design sensitivities: finite differencing, the direct differentiation method, and the adjoint variable method. Finite differencing is more or less an external loop around the multibody analysis. This means that the computational cost is proportional to the number of design variables. This also accounts for the direct differentiation method, which requires the solution of additional sensitivity equations for each design variable. However, direct differentiation is usually more accurate. Furthermore, it is often much cheaper than finite differencing when numerical analysis and sensitivity analysis can be combined in a computationally efficient way. The computational cost of the adjoint method is directly related to the number of response sensitivities that have to be calculated. The adjoint equations arise independent of the number of design variables.

If a multibody analysis program is to be extended with a design sensitivity analysis either the direct or the adjoint method should be implemented. Finite differencing may suffer from inaccuracies and problems with step size selection. Therefore, it should only be used if, from a practical point of view, no other option is available. The choice between the direct and the adjoint method is less straightforward. Additionally to the number of design variables, responses and computer processors, one also has to consider the number of load cases. In many engineering applications the multibody system design is subjected to multiple load cases. The computational cost of the direct differentiation approach is, besides the number of design variables, proportional to the number of load cases as well. This in contrary to the adjoint method that can compute sensitivities just for selected responses of each load case.

The approximate optimization process as proposed in Chapter 3 has consequences for the design sensitivity analysis of multibody systems. Two main elements are the constraint deletion, and the intermediate design variables and intermediate response quantities. The constraint deletion marks the responses that are included in the approximate optimization problem. The intermediate variables and responses allow the design engineer to improve the approximations by introducing nonlinear functional behavior that is explicitly known. Both aspects influence the design sensitivity analysis: the constraint screening limits the number of sensitivities that is actually needed; the intermediate responses suggest to calculate the sensitivities not directly for objective function and constraints, but for the more fundamental multibody responses instead such as positions, velocities, accelerations and Lagrange multipliers.

Suppose a kinematic or dynamic analysis is carried out, and the constraint screening reveals that the design sensitivities are needed for only part of the multibody responses at a fraction of all time points. For kinematically driven systems, direct differentiation is able to calculate the sensitivities at the marked time points, but cannot distinguish between the individual generalized coordinates. The adjoint variable method, however, can. For dynamically driven systems, differential or differential-algebraic equations occur. As a consequence, sensitivity equations obtained by direct differentiation have to be integrated forward in time to compute the derivatives at a specific time point. Now only the adjoint method benefits from the constraint screening, since the adjoint equations can be assembled for each time point response separately.

Roughly speaking, for one computer processor the adjoint method is favorable if the number of responses for which sensitivities are needed is substantially smaller than the number of design variables times the number of load cases. Therefore, the adjoint method requires the number of responses to be minimized. Preferably, the constraint screening would mark only one time point for each critical or potentially critical local maximum of a time dependent response. This is in contradiction with Chapter 3, where multiple points at a local maximum appeared to be beneficial to avoid oscillations if the approximations lack correct curvature. Possibly, an approximation concept with adjustable conservativeness may be able to overcome this difficulty.

Integral type of functions limit the number of objective function and constraints in the optimization problem, which is advantageous for the adjoint variable method. However, in Section 2.4 it has already been remarked that equivalent integral functions of time dependent constraints may raise numerical difficulties during the optimization. Apart from that, the combination with the adjoint variable method hinders the utilization of intermediate response variables. Take for example the inte-

gral response function:

$$\bar{r} = \int_{t_0}^{t_f} p(\mathbf{q}, t) dt. \quad (4.149)$$

Any nonlinearity present in p returns in the sensitivity:

$$\frac{d\bar{r}}{db} = \int_{t_0}^{t_f} p_{\mathbf{q}} \frac{d\mathbf{q}}{db} dt \quad (4.150)$$

via the term $p_{\mathbf{q}}$. The adjoint method treats $p_{\mathbf{q}} \frac{d\mathbf{q}}{db}$ as one scalar term, and actually never calculates the vector product itself. As a consequence, explicit nonlinear behavior embedded in p is lost since it is accumulated in $p_{\mathbf{q}}$. It cannot be used anymore to improve the approximations. This is contrary to the direct method, where $\frac{d\mathbf{q}}{db}$ is explicitly calculated and multiplied by $p_{\mathbf{q}}$ to obtain the sensitivity values.

Summarizing, design sensitivity analysis by means of direct differentiation is probably the most appropriate approach if approximate optimization with time discretization is applied. It is most suited to utilize intermediate response variables. Furthermore, using multiple time points at a local maximum, the total number of critical and potentially critical time points will often be too large to benefit from the adjoint method, even for multiple load cases. Only if the number of objective function and constraints is small and sufficient disk space is available for data storage, the adjoint method may be preferable.

Chapter 5

Stress Constrained Design of a Four-Bar Mechanism

5.1 Introduction

The stress constrained four-bar mechanism is a nice (academic) test problem to illustrate the effect of intermediate design variables and intermediate response quantities. Additionally, the influence of dynamics on the design optimization can be studied. Originally, the four-bar mechanism was used for optimization purpose by Sohoni and Haug (1982) in the context of kinematically driven multibody systems. In this chapter, the mechanism is optimized starting from a dynamic analysis. So, vibrations can be present, induced by the motion of the system.

Figure 5.1 shows the four-bar mechanism. It consists of three solid but flexible links, connected to each other and the ground by revolute joints. The three mobile links have a constant circular cross section, and a mass density of $2757 \text{ kg}\cdot\text{m}^{-3}$. The lengths of the bars are $l_1 = 0.3048 \text{ m}$, $l_2 = 0.9144 \text{ m}$, $l_3 = 0.762 \text{ m}$, and $l_4 = 0.9144 \text{ m}$, respectively. The input crank rotates at a constant angular velocity of $10\pi \text{ rad}\cdot\text{s}^{-1}$. Due to the motion, bending stresses occur in the mobile links. Stresses arising from axial forces are assumed to be negligible. The optimum design problem is now to minimize the mass of the mechanism by varying the cross sectional areas, with the bending stresses constrained to a maximum of $\sigma_a = 2.758 \cdot 10^7 \text{ Pa}$.

Each link is modeled by six beam elements. The multibody analysis package MECANO (Samtech, 1994) is used to compute the bending moments M_i^k in every k -th node of link i as a function of time. The bending stresses can then be calculated

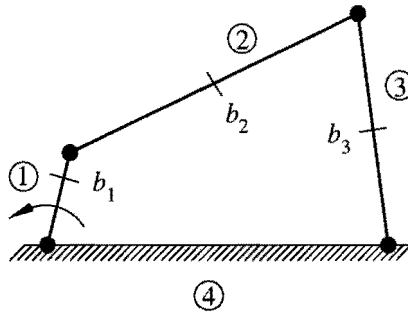


Figure 5.1: Stress constrained four-bar mechanism.

from:

$$\sigma_i^k = \frac{4\sqrt{\pi}}{A_i^{3/2}} M_i^k, \quad (5.1)$$

with A_i the cross sectional area of body i . This relation suggests to use the bending moments as intermediate response quantities, as mentioned in Section 3.3. A time interval of 0.3 to 0.5 s is considered, which exactly covers one period of steady state motion after the transient has died away.

MECANO has been coupled with the design optimization tool described in Chapter 3, using a local approximation concept. However, MECANO cannot calculate the required design sensitivities. Therefore, the derivatives with respect to the design variables are computed by means of a forward finite difference sensitivity analysis. To begin with, the four-bar mechanism is optimized with cross sectional areas and bending moments as intermediate design variables and responses. Vibrations appear to affect the optimization process. Their influence will be diminished by increasing the original Young's modulus value of the mobile links from $6.895 \cdot 10^{10}$ Pa to $6.895 \cdot 10^{11}$ Pa (imaginary experiment). For these two cases, the effect of the intermediate variables and responses will be visualized.

5.2 Design Optimization

The optimization problem of the four-bar mechanism is formulated as follows: find the diameters of the mobile links $b_i = d_i$ ($i = 1, 2, 3$) that will minimize the total mass:

$$F(\mathbf{b}) = \rho(l_1 A_1 + l_2 A_2 + l_3 A_3), \quad (5.2)$$

subject to the stress constraints:

$$\begin{aligned} \sigma_i^k(\mathbf{b}, t) &\leq \sigma_a & \forall t \in [0.3, 0.5], \\ & & i = 1, 2, 3, \quad k = 1, \dots, 7, \end{aligned} \quad (5.3)$$

in the design space: $b_i \geq 0$ ($i = 1, 2, 3$). The time interval of 0.3 to 0.5 s is discretized into 201 time points. For the approximate optimization problem, the cross sectional areas are taken as intermediate design variables:

$$b_i^I = A_i = \frac{1}{4}\pi d_i^2, \quad i = 1, 2, 3. \quad (5.4)$$

The bending moments M_i^k are used as intermediate response quantities.

The optimization is started from initial diameter values of 356.8 mm (Sohoni and Haug, 1982). The initial mechanism design has a total mass of 546 kg and is far from the maximum allowed stress. During the optimization, constraints above 50% of the constraint bound are considered as critical or potentially critical. Furthermore, constraint regionalization is applied, with two neighboring time points added before and after local maxima. The integration accuracy of a MECANO analysis is set to 10^{-6} , and the relative finite difference step size β at 10^{-3} in Equation (4.10). The initial move limit factors are taken 40% for all design variables. The move limit and convergence parameter settings correspond with Section 3.5.4.

In Figure 5.2, the optimization history of the stress constrained four-bar mechanism is visualized. The design variable values steadily decrease, and within fifteen cycles optimum design I of Table 5.1 is found. For each link one stress constraint is active. Move limits were active during the first twelve cycles. For other optimization runs optimum design II was found as well. The latter optimum equals the design mentioned in Etman *et al.* (1996b). Apparently, multiple local optima are present. Deviations with the final design of Sohoni and Haug (1982) are caused by the discretization of the mobile links and the different method of stress analysis.

A quite typical optimization history is found. During the first two cycles, none of the constraints is active or potentially active. For these cycles the maximum bending stress is indeed smaller than 50% of the stress bound. As a result, no approximation errors have been calculated. Next, the error increases from 1.8% in cycle 3 to 121% in cycle 5. Such a large approximation error is not accepted, and the search subregion of cycle 5 had to be two times reduced, until an approximate optimum design was obtained with an acceptable approximation error. This is represented by the vertical lines in Figures 5.2(a) and 5.2(b). Afterwards, the move limit factors stay at relatively small values of about 10% or smaller. Cycle 8 was repeated with halved move limits as well to keep the approximation error below 40%.

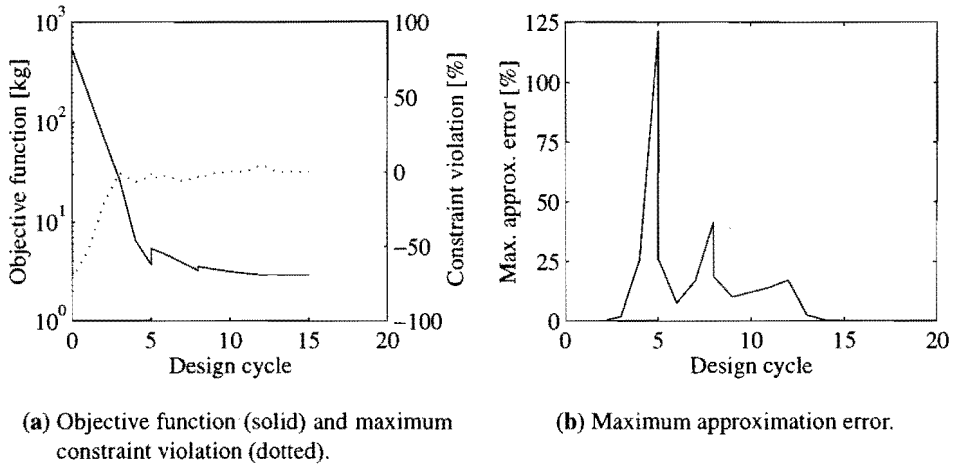


Figure 5.2: Optimization history of the stress constrained four-bar mechanism for a Young's modulus of $6.895 \cdot 10^{10}$ Pa, using cross sectional areas and bending moments as intermediate design variables and responses.

b and $F(\mathbf{b})$	Initial design	Optimum design			Optimum reported by Sohoni and Haug (1982)
		I	II	III	
d_1 [mm]	356.8	38.5	37.3	35.5	36.2
d_2 [mm]	356.8	25.2	23.6	23.0	28.1
d_3 [mm]	356.8	20.0	19.8	18.0	12.2
F [kg]	546	2.89	2.67	2.42	2.69

Table 5.1: Initial and optimum design of the stress constrained four-bar mechanism. Optimum designs I and II have been calculated for the original Young's modulus of $6.895 \cdot 10^{10}$ Pa. Optimum III corresponds with the increased Young's modulus of $6.895 \cdot 10^{11}$ Pa.

5.3 Influence of Dynamics

The cause of the move limit reductions is revealed by plotting some of the bending stresses for both initial and optimum design I. In Figure 5.3 the bending stresses at the driving node of link 1, and halfway link 2 and link 3 are plotted as a function of time. At these nodes the maximum stresses occur determining the final optimum design. Figures 5.3(a) and 5.3(b) show the bending stresses of the initial and optimum design, respectively (remark the different y-axis scaling). The initial design is clearly overweighted, while at the optimum for each link one stress constraint is active at the maximum stress bound. Additionally, the bending stresses of the optimum design show a dynamic behavior, that is (almost) absent for the initial design.

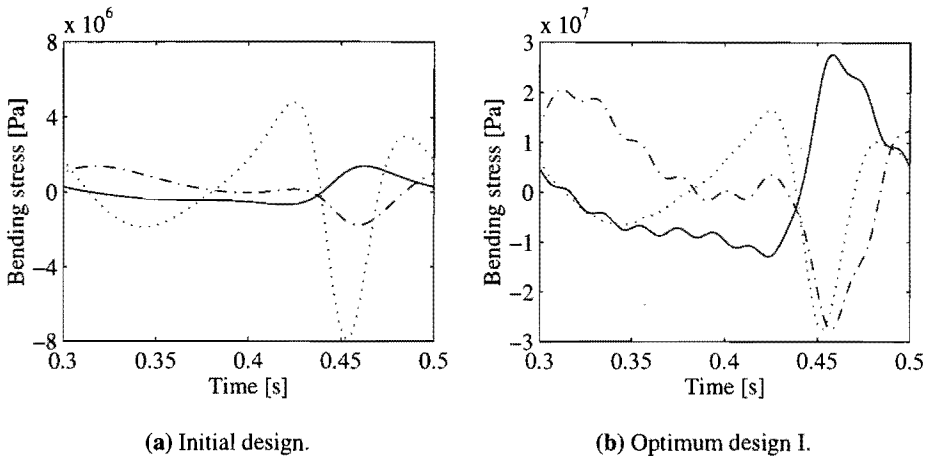
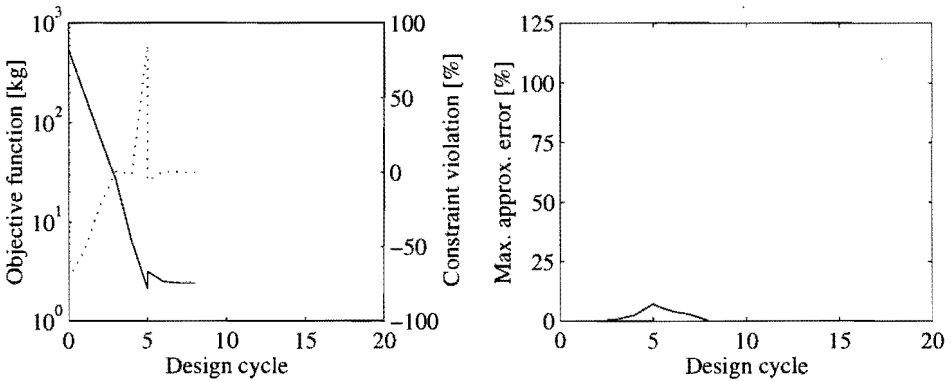


Figure 5.3: Bending stresses of the stress constrained four-bar mechanism at the driving node of link 1 (dotted), and the nodes halfway link 2 (dash dotted) and link 3 (solid), for a Young's modulus of $6.895 \cdot 10^{10}$ Pa.

The vibrations show up along with the increase of the approximation errors in the early stage of the optimization, as shown in Figure 5.2(b).

The influence of vibrations on the optimization is diminished by increasing the Young's modulus of the three mobile links. In Figure 5.4 the optimization has been repeated for a ten times magnified Young's modulus. Instead of fifteen, only eight design cycles are necessary to obtain the final optimum design. The approximation errors are very small compared with the errors obtained for the normal elasticity modulus. As a result, the move limit factors increase until a severe constraint violation occurs and the search subregion needs to be halved. For the last three cycles, the move limits are not active anymore, and convergence is automatically reached. Figure 5.5 shows the bending stresses of the corresponding initial and optimum design. Due to the small vibration amplitudes the diameters of the mobile links could be further minimized to $d_1 = 35.5$ mm, $d_2 = 23.0$ mm, and $d_3 = 18.0$, with a corresponding total mass of 2.42 kg (design III in Table 5.1).

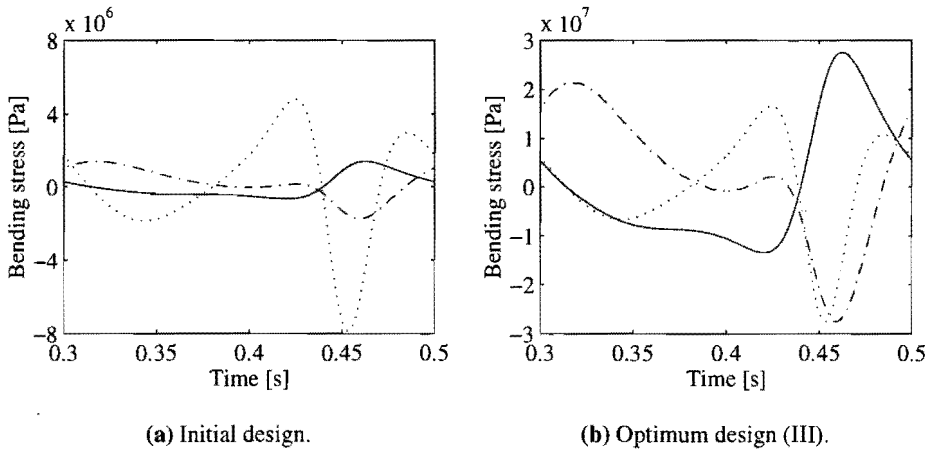
The fundamental difficulty with respect to the dynamics of the mechanism is that the vibration frequency varies if the diameters of the links are changed. According to beam theory, the frequency of vibration will decrease if the diameter of a beam becomes smaller. This effect is indeed observed for the bending stresses of the mobile links. Approximations, however, are built at fixed time points. For such a fixed time point, the true response values at the approximate optimum design are compared with the approximate responses to obtain the approximation errors in



(a) Objective function (solid) and maximum constraint violation (dotted).

(b) Maximum approximation error.

Figure 5.4: Optimization history of the stress constrained four-bar mechanism for a Young's modulus of $6.895 \cdot 10^{11}$ Pa, using cross sectional areas and bending moments as intermediate design variables and responses.



(a) Initial design.

(b) Optimum design (III).

Figure 5.5: Bending stresses of the stress constrained four-bar mechanism at the driving node of link 1 (dotted), and the nodes halfway link 2 (dash dotted) and link 3 (solid), for a Young's modulus of $6.895 \cdot 10^{11}$ Pa.

Equations (3.15) and (3.17). For large steps in the design space, the approximation errors may rise, since the approximations do not predict the shift in time. If the errors become too large, a move limit reduction is applied according to the move limit strategy of Section 3.5.4.

5.4 Effect of Intermediate Variables and Responses

The benefit of the intermediate design variables and intermediate responses defined in Section 5.2 can be confused by the dynamic behavior. To illustrate this, some additional optimizations have been carried out without any intermediate variables or responses. The original problem formulation of Sohoni and Haug (1982) is used, with cross sectional areas as design variables. The bending stresses are linearly approximated with respect to the cross sectional areas. A sequence of linear programming problems follows.

Again the two cases of normal and increased Young's modulus are studied. Firstly, mobile links with the original value of elasticity are considered. Optimum design II is obtained after nineteen cycles, of which four have been repeated with reduced search subregion. The optimization history is plotted in Figure 5.6. The corresponding case with intermediate design variables and responses required fifteen cycles, which is only slightly better (compare Figures 5.2 and 5.6). So, the effect of the intermediate design variables and intermediate responses seems to be marginal. The same problem was noticed in Etman *et al.* (1996b), who expected a significant improvement in comparison with sequential linear programming reported in Etman *et al.* (1994), but did not find it.

Next, the Young's modulus is increased towards $6.895 \cdot 10^{11}$ Pa, which reduces the contribution of the dynamic behavior. Now the optimization converges towards the same optimum design that was calculated when using intermediate variables and responses. However, fourteen cycles are necessary instead of eight. Furthermore, the errors of Figure 5.7 are larger than the errors of Figure 5.4. These observations confirm that the intermediate design variables and intermediate response quantities do yield better approximations.

5.5 Conclusion and Discussion

Intermediate design variables and intermediate response quantities can improve the approximate optimization problem. For the stress constrained four-bar mechanism the selected cross sectional areas and bending moments enhance the optimization process, alike the analytical examples of Chapter 3. However, the improvement

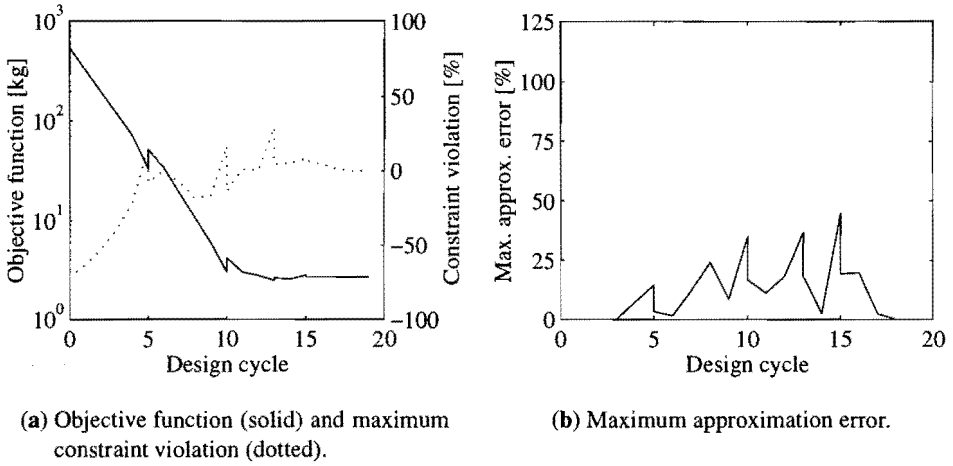


Figure 5.6: Optimization history of the stress constrained four-bar mechanism for a Young's modulus of $6.895 \cdot 10^{10}$ Pa, without intermediate variables and responses.

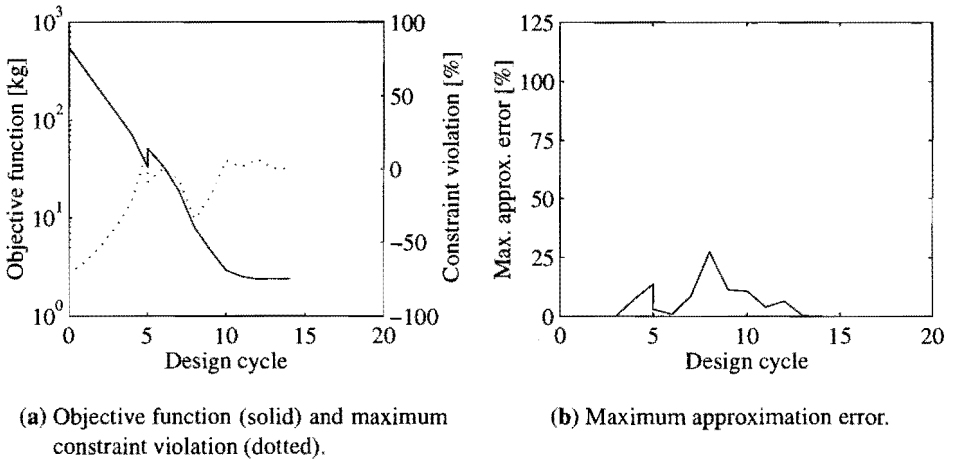


Figure 5.7: Optimization history of the stress constrained four-bar mechanism for a Young's modulus of $6.895 \cdot 10^{11}$ Pa, without intermediate variables and responses.

clearly differs for the original and the increased Young's modulus. In the latter case, the vibrations are far less outstanding, and the approximation errors can be much more suppressed, yielding a fast convergence of the optimization. The mechanism with original Young's modulus shows much more difficulties. The vibrations have such an amplitude that a shift of the vibration frequency affects the quality of the approximations adversely.

Possibly, new intermediate design variables and intermediate responses may be defined that include the change of dynamic behavior. A linear dependency between the lowest eigenfrequencies of a link and the corresponding diameter seems to be present. The mass effect of one link on the bending stresses of the other links is less trivial. It is worthwhile to investigate whether effective intermediate variables and responses can be identified, not just because of the mechanism itself, but mainly to gain insight in a broader perspective of multibody systems.

Besides, the improved behavior for excluded dynamics suggests to divide the optimization of mechanisms with flexible bodies into two phases. During the first phase the optimization is carried out for rigid bodies. Next, the calculated optimum design can be used as a high quality initial design for an optimization with flexible bodies. The new optimum will probably be only slightly different. For the stress constrained four-bar mechanism design III of Table 5.1 lies fairly close to designs I and II.

Chapter 6

Optimum Crashworthiness Design of a Vehicle Restraint System*

6.1 Introduction

Safety measures like airbags and safety belts are nowadays commonly applied to improve the crashworthiness of road vehicles. Many governments stimulate or oblige by law car producers to include these devices in the vehicle design. Maximum values are imposed on parameters that quantitatively describe the severity of injuries. Examples are the Head Injury Criterion (HIC) and the chest 3 MilliSeconds criterion (chest 3MS). Physical crash tests of newly designed vehicles are used to determine the vehicle's crashworthiness. However, these physical tests are very cost expensive, and therefore car producers generally use crash simulation software in the design stage instead.

Computer crash simulation enables numerical design optimization. However, only a few applications of design optimization to crashworthiness analysis have been published. Due to the impact situation usually large displacements and extreme material deformations are present. Furthermore, contacts may occur in the deforming material and between occupant and vehicle interior (e.g. airbag or steering wheel). As a result, the behavior of injury parameters as a function of the design variables may be highly nonlinear or even discontinuous, especially for complicated crash models. This hinders the numerical optimization.

*This chapter has been partly reproduced from Etman *et al.* (1996a). The research was carried out in close cooperation with TNO Road-Vehicles Research Institute, Delft (see also Section 3.6).

In literature two different optimization approaches can be identified for crashworthiness optimum design. The first approach is to model the vehicle and restraint system by means of a simplified mechanical model such that a smooth optimization problem is obtained. Dias and Pereira (1994) can be classified in this category. The second approach starts from a complex, usually finite element, crash model and an optimization strategy that can handle non-smooth functional behavior. Bennett and Park (1995), for example, used sequential linear programming with somewhat larger finite difference steps to calculate the sensitivities, as well as Vanderplaats' multi-point optimization method (Vanderplaats, 1979).

In this chapter the multi-point approximate optimization tool of Section 3.6 is applied to the crash simulation package MADYMO (TNO, 1994). MADYMO is a combined multibody finite element program, which has been developed at TNO Crash-Safety Research Center in The Netherlands. Large-scale MADYMO simulations indeed show a noisy functional behavior of the injury criteria. Moreover, these crash analyses are often computationally expensive. It is expected that a sequence of multi-point approximations can solve the noisy crash optimization problem in a computationally efficient way.

Firstly, the crashworthiness optimization problem is described in more detail, and some important injury criteria are introduced. Next, a short overview is given of other strategies that were previously used for design optimization using MADYMO. Within this context the multi-point approach is explained. Finally, the vehicle restraint system is presented that will be optimized using the multi-point approximate optimization tool. Optimizations are carried out for both two and six design variables. In both cases good convergence is found.

6.2 Crashworthiness Optimization Problem

Crashworthiness design is characterized by a variety of objective functions, constraints and design variables. Typical crashworthiness responses are occupant injury criteria, as well as structural displacements and accelerations. For example, the objective may be to minimize a weighted composite injury criterion subject to constraints on individual injury criteria and structural constraints. An other option is to select the total mass or some production cost as objective function. Design variables are directly related to the specific restraint system. For an airbag one can think of variables defining the geometry, or variables related to the gas flow. Probably needless to say, but the choice for design variables, objective function and constraints should be carefully considered, since it has a large influence on the design optimization results.

A well-known and commonly applied injury parameter is the Head Injury Criterion (HIC). It is a measure for the severity of the linear (i.e. no rotational) acceleration on the human head, and is quantitatively formulated as (see Hardy *et al.*, 1994, for a literature review of injury biomechanics):

$$HIC = \max_{t_0 \leq t_1 \leq t_2 \leq t_f} (t_2 - t_1) \left[\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} a(t) dt \right]^{2.5}. \quad (6.1)$$

Herein, t_0 and t_f are the start and final time of the numerical simulation, respectively. The response $a(t)$ is the linear head acceleration in g, measured at the head's center of gravity. A time interval $[t_1, t_2]$ is searched for that will maximize the integral function of the acceleration. For the HIC, this time interval is restricted to a maximum of 36 milliseconds.

Several other injury parameters are used as well. The 3 MilliSeconds criterion (3MS) is defined as the highest acceleration level with a duration of at least 3 milliseconds. It can be applied to several parts of the human body, like the head and the chest. Examples of injury criteria that are not related to accelerations are the chest deflection criterion and the maximum femur compression load. MADYMO can automatically calculate the values of these injury criteria for any impact situation.

For large-scale MADYMO crash simulations the behavior of the injury parameters as a function of the design variables may show a band of noise added to the global response. This is illustrated by a crash simulation example of a full-scale frontal impact. The interaction between occupant and airbag is investigated by considering a range of vent diameters of 30 to 70 millimeters. In Figure 6.1 the calculated HIC values have been plotted as a function of the vent diameter. Clearly, an optimum diameter is present. However, the noise is quite disturbing, and if it is present for one or more injury parameters, a sensitivity based optimization algorithm will probably fail. Therefore, the optimization strategy to be selected should be able to handle this kind of non-smooth behaving design problems using as least numerical crash simulations as possible.

6.3 Optimization using MADYMO

Approximate optimization strategies are able to overcome the difficulties encountered in optimum crashworthiness design. Especially, multi-point approximations that do not use sensitivities are suitable to deal with both the computationally expensive numerical analysis and the non-smooth optimization problem. The basic principle is to generate approximations of objective function and constraints in a certain part of the design space, and to solve the optimum point for this approxi-

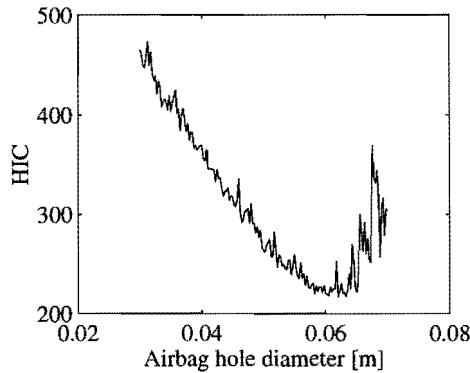


Figure 6.1: Noise on global HIC response.

mate optimization problem. The approximate optimization problem is smooth and explicitly known, and can be easily solved using a standard mathematical programming algorithm.

In Section 3.2 a classification of approximation concepts has been given. Two basic multi-point approximation concepts can be distinguished: global and mid-range approximations. Using the global concept, approximation models of objective function and constraints are built that create an explicitly known approximate optimization problem in the complete design space or a large part of it. To generate global approximations, often response-surface techniques are used (Schoofs, 1987). Response-surface model building starts with postulating the approximate model functions. Then, an experimental design is selected defining the set of the design points for which computer experiments are carried out. Finally, regression analysis is used to estimate the unknown parameters of the approximate models by fitting the numerical response data.

Both Bosio and Lupker (1991) and Schoofs *et al.* (1992) constructed global response-surface models for MADYMO crashworthiness design problems. Bosio and Lupker (1991) used Taguchi's method to build a global linear response-surface model including interaction terms, and studied the influence of several seat belt and airbag design variables on the HIC in a frontal crash situation. Schoofs *et al.* (1992) constructed linear and quadratic response-surface models from a set of MADYMO analyses. An objective function built from several neck injury parameters was formulated to optimize a child seat.

However, response-surface model building is a highly user-interactive and iterative process. The selection of appropriate model functions is often rather difficult. Additionally, the number of analyses increases exponentially for a growing number

of design variables or more elaborate model functions. These disadvantages are not present for multi-point mid-range approximations. The region in which mid-range approximations are built is restricted. The model functions can remain simple, and, as a consequence, the number of analysis points in the experimental design will be much smaller. The final optimum solution has to be obtained from a sequence of design cycles of approximate model building and optimization.

Instead of global response-surface modeling, the multi-point approximate optimization tool of Section 3.6 has been coupled with MADYMO for optimization purpose. Linear approximations of objective function and constraints are generated, yielding a multi-point sequential linear programming process. Move limit and convergence parameters have to be set according to the noise amplitude that is expected to be present. The impact-absorber example of Section 3.7.1 showed that an under estimation may lead to oscillations, and an over estimation to a premature convergence. Goualou *et al.* (1996) also developed an optimization tool for MADYMO. However, they did not use approximation concepts, but made a direct coupling with the Nelder and Mead simplex algorithm, that is rather robust for discontinuities since it does not use gradients.

6.4 Restraint System for Frontal Impact

The optimization tool is used to design a combined airbag and belt restraint system for optimum crash safety. The driver side of a typical sedan is modeled for frontal collision at $50 \text{ km}\cdot\text{h}^{-1}$ using the multibody and finite element capabilities of MADYMO (see Figure 6.2). Key elements are a collapsible steering column, a seat, a three-point belt system, an airbag, and a dummy. The dummy represents an average adult male occupant. Initially, the airbag is stowed away in the 14 by 10 centimeter container on the steering wheel. At impact, the airbag is inflated and a retractor mechanism pretensions the belt. Airbag and belt should reduce head and chest accelerations and prevent contact with the steering wheel.

Six design variables have been assigned. For the airbag restraint system the airbag diameter, the airbag hole diameter, and the inflator gas mass are considered. Increasing the airbag diameter means that a larger part of the body of the occupant is covered. Though, at the same time, the airbag will become much softer for constant inflator gas mass. The airbag hole diameter directly influences the damping. For a small hole hardly any damping is present: the airbag behaves more or less like a pure elastic ball. Design variables of the belt restraint system are the belt webbing, the belt load limiter and the pretensioner distance. The webbing determines the stiffness of the belt, whereas the load limiter confines the forces imposed upon the

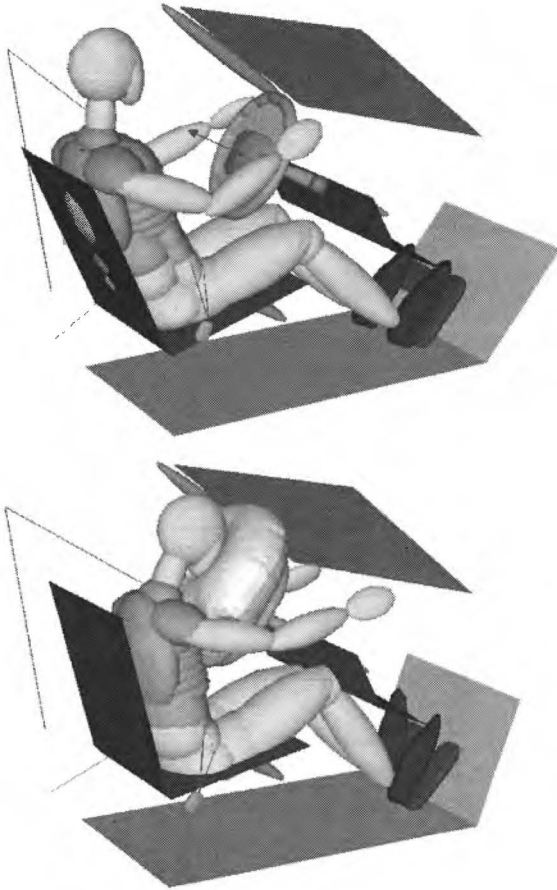


Figure 6.2: MADYMO restraint system model for frontal impact. The situations before and 75 milliseconds after impact are plotted.

chest.

The optimization problem is to find the design variable values that will minimize the head injury criterion subject to injury constraints on head, chest, and shoulder. For the head, the 3MS acceleration criterion should not exceed 50 g. Furthermore, contact with the steering wheel should be avoided. This is represented by a constraint on the distance between head and hub of the steering wheel. The maximum chest 3MS acceleration and chest deflection are 35 g and 42.4 mm, respectively. Finally, the shoulder belt load is constrained to a maximum of 7 kN. These constraint bounds are more restrictive than the demands made by most governments. The lower and upper bounds of the design variables are given in Table 6.1. Objective function and constraints are summarized in Table 6.2.

b_i	Design variable	b_i^l	b_i^u
b_1	Airbag diameter [m]	0.5	0.9
b_2	Airbag hole diameter [m]	0.03	0.07
b_3	Inflator gas mass [kg]	0.02	0.06
b_4	Belt webbing [% strain at 10 kN]	0.03	0.18
b_5	Pretensioner distance [m]	0.0	0.15
b_6	Belt load limiter [kN]	2.0	8.0

Table 6.1: Design variables and design space of the restraint system.

F, g	Response	Bound
F	HIC	-
g_1	Head-hub distance [m]	> 0
g_2	Head 3MS [g]	≤ 50
g_3	Chest 3MS [g]	≤ 35
g_4	Belt load [kN]	≤ 7
g_5	Chest deflection [mm]	≤ 42.5

Table 6.2: Objective function and constraints of the restraint system.

6.5 Optimization Results

To begin with, just two design variables are studied in the optimization: the airbag diameter b_1 and the airbag hole diameter b_2 . All other variables are fixed at a constant value corresponding to the initial design given in Table 6.3. A grid of design points has been analyzed to visualize the optimization problem. Figure 6.3(a) shows the different constraints together with the feasible domain. The noisy functional behavior is clearly present. Contours of the objective function are plotted in

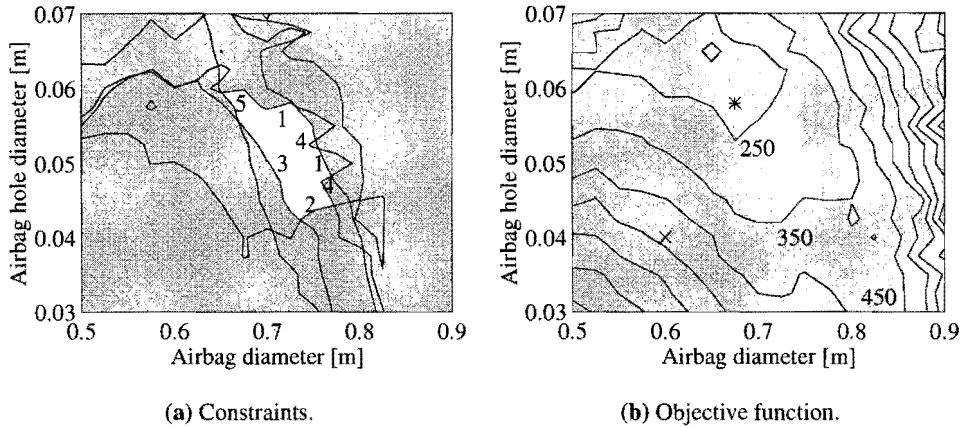


Figure 6.3: Two dimensional restraint optimization problem. The feasible domain is white. Initial and calculated optimum design are marked with \times and $*$, respectively.

Figure 6.3(b), together with the feasible domain. The optimum region is near the upper left part of the feasible area. This region is bounded by constraints g_1 , g_3 and g_5 .

The optimization is started from $b_1 = 0.6$ m and $b_2 = 0.04$ m, with search direction $\mathbf{d}^{(1)} = [1, 1]$, and extrapolation factor $\alpha = 0.1$. The initial search subregion dimensions $s_i^{(1)}$ are set to 20% of the design space $b_i^u - b_i^l$, and the noise is estimated at about 5%. Starting from the infeasible initial design, the optimum is found at $b_1 = 0.684$ m and $b_2 = 0.06$ m after four cycles (13 MADYMO analyses). This optimum design corresponds quite well with Figure 6.3(b). The final objective function value is 218.

Next, all design variables are included. The initial design of Table 6.3 is used. Extrapolation factor and initial subregion size stay the same. The start search direction is again forward for all design variables. An optimum design is obtained after seven cycles. Note that both airbag diameter and airbag hole diameter converge towards the optimum of the two-design-variables case. The objective function values of 206 is comparable as well. The design variables b_3 , b_4 and b_6 hardly change, which may indicate that their initial values are already near the optimum solution.

To investigate this, the sequential approximate optimization is re-started at the optimum design of cycle 7. Both move limit values and start search direction are re-set at the initial values of the first run. Now, the inflator gas mass b_3 starts to increase significantly, together with an additional increase of the airbag diameter and the airbag hole diameter. So the loss of stiffness and damping due to changes in b_1 and b_2 is compensated. During the second optimization run the objective func-

Design variable	Initial design	Optimum first run	Optimum second run
b_1	0.6	0.673	0.753
b_2	0.04	0.0604	0.0690
b_3	0.04	0.0381	0.0568
b_4	0.09	0.0753	0.0703
b_5	0.05	0.0847	0.0796
b_6	4.0	4.0	3.6
F	533	206	111
g	g_2, g_3, g_5 violated	g_1, g_3, g_5 active	g_1 active
Cycles		7	8

Table 6.3: Initial and optimum design of the restraint system.

tion is minimized towards a HIC of 111. Only constraint g_1 is active. The design variables corresponding to the belt restraint system stay near their start positions in the design space.

6.6 Conclusion and Discussion

Design optimization using MADYMO is complicated by a band of noise present on the global response of objective function and constraints. An optimization method is necessary that does not use design sensitivities. Some references showed that good results were obtained by global response-surface model building. However, this approach is highly user-interactive and becomes impractical for increasing number of design variables. A more manageable approach follows with multi-point mid-range approximations. The multi-point approximate optimization tool has successfully solved optimization problems that combine non-smooth functional behavior with a computationally expensive numerical analysis.

For the restraint system a good convergence was found. The objective function value could be significantly reduced. Even more important, the calculated optimum design satisfies all imposed constraints, in contrary to the initial design that violates three constraints. The selected move limit and convergence parameters performed reasonably well, although the premature convergence of the optimization during the first run may indicate that the noise is actually less than expected. Probably this is not the case, but multiple local optima are present. For multi dimensions it is difficult to know which of both cases is true. Some experimentation and engineering experience is required to properly select the move limit parameters.

Furthermore, the restraint problem showed that the design variables cannot be treated independently. Loss of damping due to an increase of the airbag hole diameter can be partly compensated by an increase of the inflator gas mass. For large number of design variables it is really difficult to survey the individual effects, not to mention the interaction effects. Fortunately, the optimization tool takes this work off the designer's hands, so that he can concentrate on the interpretation of the optimization results.

The optimizations of the restraint system have been carried out using multi-point sequential linear programming. In Chapter 3 it has been remarked that both single-point and multi-point approximations can be improved by inclusion of suitable intermediate design variables and responses. Lust (1992), for example, showed that for structural crashworthiness constraints, such as the vehicle crash severity index, intermediate design variables and responses can be defined that yield more robust constraint approximations. Probably many more can be found, reflecting the mathematical and physical background of the crashworthiness responses.

Chapter 7

Design of a Stroke Dependent Damper for the Front Axle Suspension of a Truck*

7.1 Introduction

Comfort is becoming increasingly important in truck design. Truck designers tend to lower the stiffness of the axle suspensions for the benefit of comfort of both driver and cargo. Stiffness reduction, however, comes at the expense of increased suspension deflections. This is not favorable since an increase of the suspension working space means a decrease of the available payload volume. Therefore, several advanced suspension systems have been introduced during the last decade to improve the compromise between conflicting measures such as comfort and suspension deflection working space (see Elbeheiry *et al.*, 1995, for a classified bibliography). Well-known examples are semi-active and active suspensions. A semi-active suspension can rapidly adjust its settings (usually a damping characteristic), whereas an active suspension can both dissipate and supply energy by means of an actuator.

Reduction of extreme suspension deflections may be also obtained by inclusion of a stroke dependent damper in the axle suspensions. This is a very simple suspension system that may prove to be a cost-effective alternative for semi-active or active suspensions. For normal operating conditions only small suspension deflections will occur. However, when the truck comes across an incidental big road

*The results presented in this chapter were obtained during the master's thesis work of Vermeulen (1996) at Eindhoven University of Technology. The research was carried out in close cooperation with DAF Trucks N.V., Eindhoven.

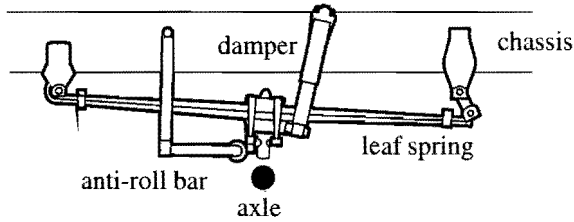


Figure 7.1: Schematic representation of the front axle suspension of a truck.

disturbance, like a pothole or a traffic hump, extra damping may be applied to compensate for the negative effects of a low stiff suspension.

Generally, the basic axle suspension consists of a spring, a damper, and an anti-roll bar (see Figure 7.1). Various types of dampers are commercially available. Within certain constraints, the design engineer can freely select the damper characteristic. A stroke dependent variant can, for example, be obtained by means of extra bypass channels and valves. To optimize the characteristics of such a new damper, vehicle model simulation and numerical optimization is used. The stroke dependent damping is modeled by adding an extra damper parallel to the original axle suspension, which will only contribute during large deflections. The optimization problem is then to find the damping curve of the added nonlinear damper that will give the best performance.

Some researchers already applied design optimization in the context of nonlinear damper characteristics. Demić (1989) used a modified Nelder-Mead method to optimize the parameters of elasto-damping elements in a 4-DOF (degree of freedom) vehicle model subjected to micro-roughness of asphalt road. Spentzas (1993) applied Box's method on a 7-DOF model with deterministic road irregularities alike a traffic hump. The paper of Eberhard *et al.* (1996) comes near the design problem described in the previous paragraph. However, they did not consider a stroke dependent damper, but optimized the damping curve of the original shock-absorber itself. The damping curve was parameterized by Hermite-splines, and an SQP algorithm was used to solve the optimization problem when driving over a bump. Finally, Tamis (1994) did indeed study and optimize a stroke dependent damper for the rear axle suspension of a truck. Global response-surface models were built of the maximum suspension deflection and the maximum vertical chassis acceleration as a function of four parameters defining the nonlinear damping curve.

In the current study, the design optimization tool described in Chapter 3 is used to obtain the optimum damping curve of the extra damping for the front axle suspension of a truck. Objective is to reduce the large *inward* suspension deflections. A local approximation concept is chosen with design sensitivities calculated by fi-

nite differences. At first, the truck is modeled by means of a two degree of freedom system, which is often called a quarter car model. The multibody software package DADS (CADSi, 1995) is used to predict the dynamic displacements and accelerations. Three typical road undulations are studied, leading to an optimization problem with multiple loading cases. Afterwards, a full-scale three-dimensional DADS model is used in the optimization. This model has 34 degrees of freedom and is a quite realistic representation of a DAF FT95 tractor-semitrailer combination (Bekkers, 1995).

7.2 Design Problem

Dampers with a characteristic alike Figure 7.2(a) are commonly used in vehicle axle suspensions. For the design problem considered here, the original nonlinear damper of the axle suspension is supposed to have exactly the relation between damper force F_D and relative velocity v_D as plotted in Figure 7.2(b). Although the rebound stage (outward relative velocity $v_D > 0$) differs from the compression stage (inward relative velocity $v_D < 0$), the fundamental damper force characteristics resemble. For low velocities the damping force shows a quadratic functional behavior caused by the turbulent oil flow through small orifices or between piston and cylinder. This part of the curve is called bleed. Then a spring controlled valve opens, and a transition towards a flatter part of the curve follows (blow-off). Finally, the curve may rise again quadratically at higher velocities (port) due to the turbulent flow resistance of the total valving.

The generic shape of the damping curve can, to some extent, be adapted to meet the specifications of the designer. The steepness of the quadratic parts, the moment of transition and the slope of the linear part can be realized for a range of values. The damper producer just has to change the diameters of the orifices, or adapt the spring controlled valve with respect to preload and stiffness. This can be separately done for rebound and compression stage.

The rebound and compression stage of the damping curve are parametrized using the following empirical relation:

$$F_D(v_D) = \frac{\beta_0 v_D^2 (\beta_1 + \beta_2 v_D)}{\beta_0 v_D^2 + \beta_1 + \beta_2 v_D} + \beta_3 v_D^2. \quad (7.1)$$

For small velocities this curve approaches $\beta_0 v_D^2$, assumed that $\beta_3 v_D^2$ does not yet contribute ($\beta_0 \gg \beta_3$). Increasing v_D yields the much flatter blow-off phase of the curve, provided that $\beta_0 v_D^2$ is substantially larger than $\beta_1 + \beta_2 v_D$. Finally, the magnitude of β_3 determines when $\beta_3 v_D^2$ comes into play. The solid line in Figure 7.2(b)

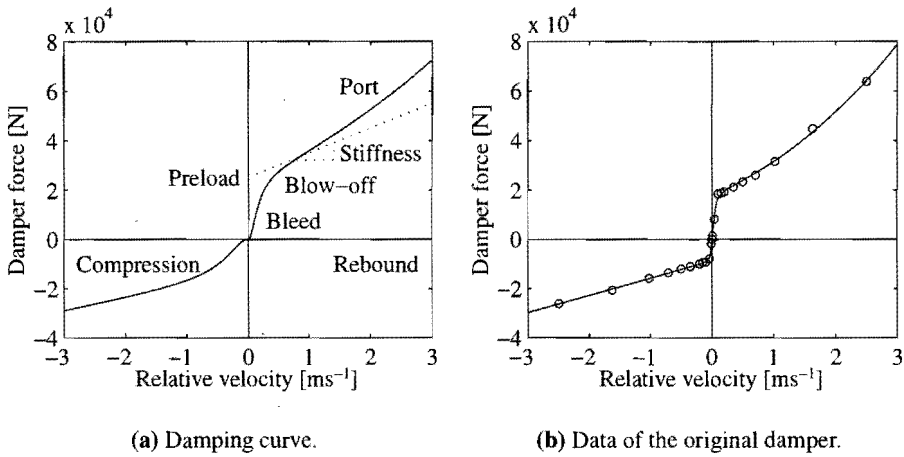


Figure 7.2: Nonlinear damper characteristic.

Parameter	Compression ($v_D < 0$)	Rebound ($v_D > 0$)
Bleed	$\beta_0^c = -26.0 \cdot 10^6 \text{ N s}^2 \text{ m}^{-2}$	$\beta_0^r = 11.2 \cdot 10^6 \text{ N s}^2 \text{ m}^{-2}$
Blow-off preload	$\beta_1^c = -8.73 \cdot 10^3 \text{ N}$	$\beta_1^r = 18.3 \cdot 10^3 \text{ N}$
Blow-off stiffness	$\beta_2^c = 7.11 \cdot 10^3 \text{ N s m}^{-1}$	$\beta_2^r = 9.62 \cdot 10^3 \text{ N s m}^{-1}$
Port	$\beta_3^c = 0.0276 \cdot 10^3 \text{ N s}^2 \text{ m}^{-2}$	$\beta_3^r = 3.52 \cdot 10^3 \text{ N s}^2 \text{ m}^{-2}$

Table 7.1: Estimated curve parameters of the original nonlinear damper.

represents the fit of (7.1) onto the damping data, which has been calculated by non-linear regression. The damping curve resembles the data remarkably well. The estimated parameters for rebound and compression stage are shown in Table 7.1.

Under normal driving conditions the suspension deflections remain small and the extra damping is not needed. To this end, the stroke dependent damping does not contribute until the inward suspension deflection (compression) comes above 0.04 m, and linearly increases towards full damping force at 0.05 m. For compressions larger than 0.05 m the additional stroke dependent damping force directly follows from the nonlinear damping curve. The objective is to find the most effective stroke dependent damping curve.

The optimization problem is to minimize the maximum inward suspension deflection subject to constraints on the vertical acceleration of the chassis for three road undulations. The design variables correspond with the parameters of the damping curve: bleed $b_1 = -\beta_0^c$, blow-off preload $b_2 = -\beta_1^c$, and blow-off stiffness $b_3 = \beta_2^c$ for compression, and bleed $b_4 = \beta_0^r$, blow-off preload $b_5 = \beta_1^r$, and blow-off stiffness $b_6 = \beta_2^r$ for rebound. They are summarized in Table 7.2, together with

the un-scaled lower and upper bounds. Parameters β_3^c and β_3^r are not included as design variables, but kept fixed at the values of Table 7.1. Preliminary calculations showed that their contribution is small in the operating range.

Design variables	Lower bounds	Upper bounds
$b_1 = -\beta_0^c$ [$10^6 \text{ N s}^2 \text{ m}^{-2}$]	0.3	22
$b_2 = -\beta_1^c$ [10^3 N]	2	200
$b_3 = \beta_2^c$ [10^3 N s m^{-1}]	0.7	70
$b_4 = \beta_0^r$ [$10^6 \text{ N s}^2 \text{ m}^{-2}$]	0.3	22
$b_5 = \beta_1^r$ [10^3 N]	2	200
$b_6 = \beta_2^r$ [10^3 N s m^{-1}]	0.7	70

Table 7.2: Design variables with lower and upper bounds.

The following incidental road undulations have been selected: a traffic hump, a wave and a railway crossing (see Figure 7.3). The traffic hump is crossed at a speed of $15 \text{ km} \cdot \text{h}^{-1}$. It lasts 4.25 m and has a maximum height of 0.25 m for 2.25 m of the total width. The wave corresponds with a squared sinus of 25 m long and 0.5 m high. The corresponding driving speed is $80 \text{ km} \cdot \text{h}^{-1}$. Finally, the third road profile is represented by data measured at a railway crossing somewhere near Eindhoven. It is crossed at a speed of $25 \text{ km} \cdot \text{h}^{-1}$. Both traffic hump and railway crossing are built from piece-wise linear interpolations, with a slight rounding of the corners using digital filtering techniques.

7.3 Quarter Car Model

7.3.1 Problem Description

A very simple model of the front side of the truck is the two-DOF system as depicted in Figure 7.4(a). The sprung mass (chassis) and unsprung mass (wheel axle) are denoted by m_c and m_a , respectively. The corresponding vertical displacements are y_c and y_a . The tire is modeled by a linear spring with stiffness k_t . In vertical direction it is 'attached' to the road which means that tire lift off cannot occur. The axle suspension consists of a linear spring k_s and a nonlinear damper c_s with a damping characteristic as plotted in Figure 7.2(b). The mass and stiffness parameter values are given in the table next to the quarter car model.

Numerical analysis of this two-DOF system using DADS (CADSi, 1995) yields the displacement and acceleration responses presented in Figure 7.5. The three road undulations give rise to quite different responses. Compared with the traffic

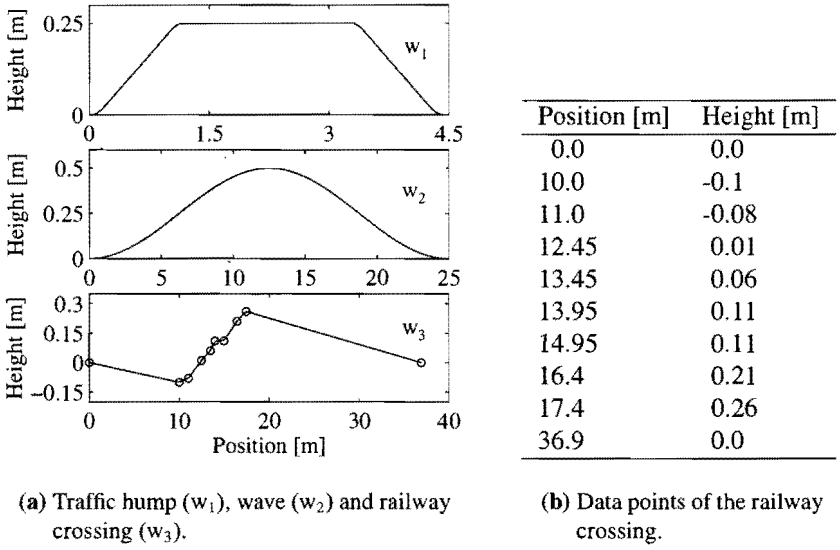


Figure 7.3: Road undulations.

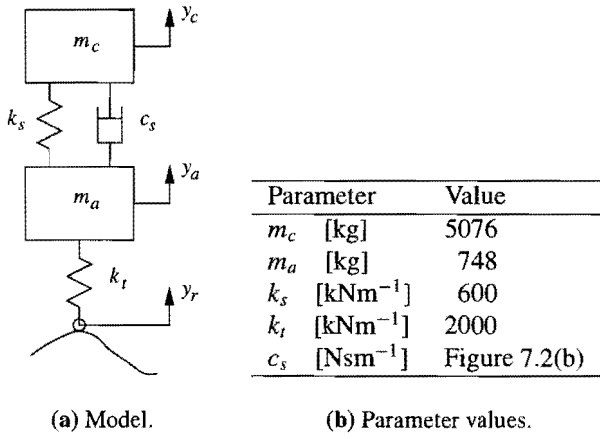


Figure 7.4: Quarter car model.

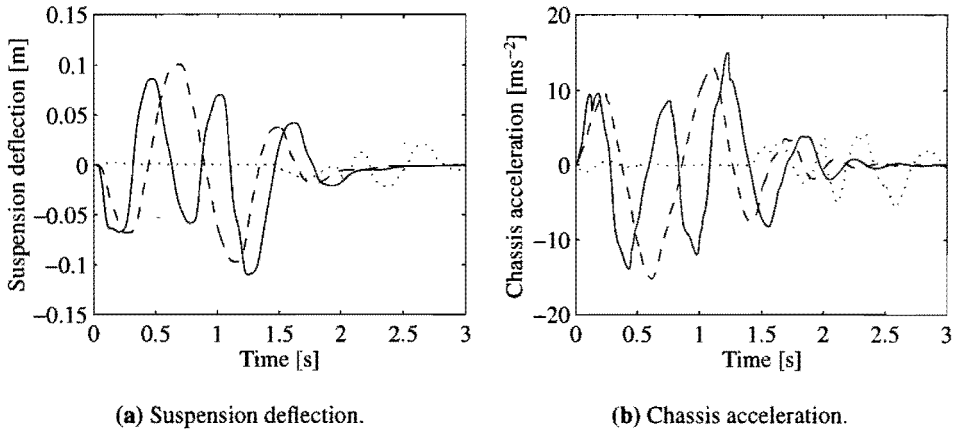


Figure 7.5: Displacement and acceleration responses of the quarter car model for the traffic hump (solid), wave (dashed), and railway crossing (dotted).

hump and the wave, the railway crossing is rather innocent: the amplitudes of both deflection and acceleration are much smaller. The maximum negative suspension deflection is 0.11 m, and occurs for the traffic hump after 1.25 s. The maximum acceleration is near 15 ms^{-2} , reached for both the traffic hump and the wave at different moments in time.

The stroke dependent damper is placed parallel to spring k_s and damper c_s of the original axle suspension. Now, the optimization problem is to determine the parameters \mathbf{b} describing the curve of the extra damper that will minimize the maximum inward (i.e. negative) suspension deflection :

$$F(\mathbf{b}) = \max_{t \in [0, 3]} -\{y_c(\mathbf{b}, t) - y_a(\mathbf{b}, t)\} \quad (7.2)$$

subject to a maximum chassis acceleration a_c^m of 20 ms^{-2} :

$$g_1(\mathbf{b}, t) = \frac{|\ddot{y}_c(\mathbf{b}, t)|}{a_c^m} \leq 1 \quad \forall t \in [0, 3], \quad (7.3)$$

a maximum outward (i.e. positive) suspension deflection d_{so}^m of 0.14 m:

$$g_2(\mathbf{b}, t) = \frac{y_c(\mathbf{b}, t) - y_a(\mathbf{b}, t)}{d_{so}^m} \leq 1 \quad \forall t \in [0, 3], \quad (7.4)$$

and a maximum tire deflection d_t^m of 0.09 m:

$$g_3(\mathbf{b}, t) = \frac{|y_a(\mathbf{b}, t) - y_r(t)|}{d_t^m} \leq 1 \quad \forall t \in [0, 3]. \quad (7.5)$$

The corresponding side-constraints have been given in Table 7.2.

7.3.2 Optimization Results

The optimum design problem is solved using the sequential approximate optimization tool of Chapter 3. To begin with, the max-value operator is removed from the objective function by an artificial design variable, alike Section 2.3. Furthermore, every time dependent constraint is replaced by 601 time point constraints, equally distributed on the time interval of 0 to 3 s. Time point accelerations and displacements are linearly approximated with respect to the design variables. No intermediate design variables or intermediate response quantities are introduced, resulting in a series of linear programming problems. The required gradients of the displacements and accelerations are obtained by finite differencing, with a relative step size β of 10^{-3} in Equation (4.10). The integration accuracy of the DADS analysis is set to 10^{-7} . The convergence parameter values for the optimization are $\text{objacc} = 0.1\%$ and $\text{vioacc} = 0.1\%$ (see Section 3.5.4 and Appendix A).

Within eleven cycles the approximate optimization process converges, using initial move limit factors of 30% for all design variables. The optimization history is visualized in Figure 7.6. Clearly, the traffic hump is decisive. The initial and optimum design variable values are given in Table 7.3. For compression, the bleed b_1 moves towards a small value, while the preload b_2 increases to its upper bound value. The stiffness of the compression blow-off b_3 remains relatively small. The design variables of the rebound stage behave different, except for the rebound blow-off b_6 which stays at the lower bound value. The rebound bleed b_4 is set at the upper bound level, whereas the preload b_5 hardly changes. In Figure 7.7, the suspension deflection and chassis acceleration corresponding to the optimum design are plotted. They can be compared with the responses of the initial design (see Figure 7.5). The maximum negative suspension deflection has been minimized from 0.102 m to 0.0795 m with the maximum acceleration bounded to 20 ms^{-2} . The constraint on the tire deflection has never become active.

Design variable	Initial design	Optimum design
b_1 [$10^6 \text{ N s}^2 \text{ m}^{-2}$]	10.0	0.563
b_2 [10^3 N]	20.0	200
b_3 [10^3 N s m^{-1}]	0.700	1.35
b_4 [$10^6 \text{ N s}^2 \text{ m}^{-2}$]	10.0	22.0
b_5 [10^3 N]	20.0	28.6
b_6 [10^3 N s m^{-1}]	0.700	0.700

Table 7.3: Initial and optimum design of the quarter car model with stroke dependent damping for a maximum acceleration of 20 ms^{-2} .

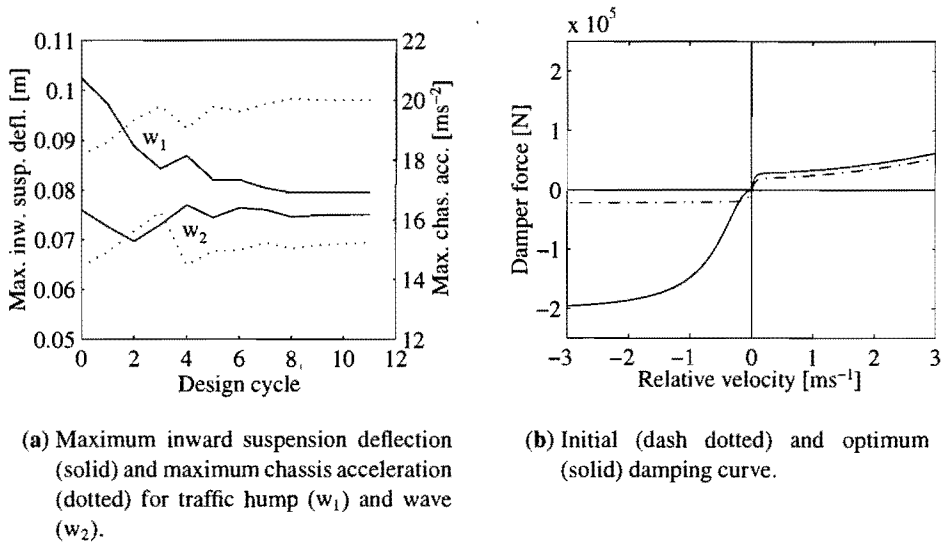


Figure 7.6: Optimization history of the quarter car model with stroke dependent damping for a maximum acceleration of 20 ms^{-2} .

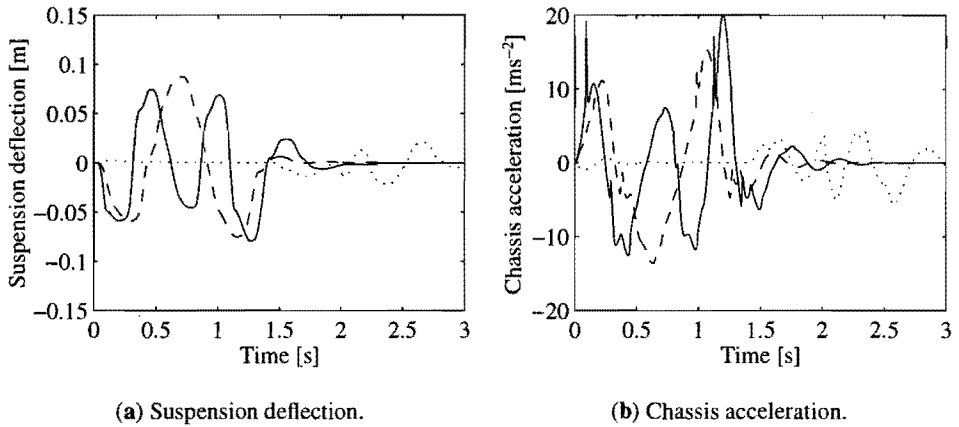


Figure 7.7: Displacement and acceleration responses of the quarter car model with optimized stroke dependent damping for the traffic hump (solid), wave (dashed), and railway crossing (dotted).

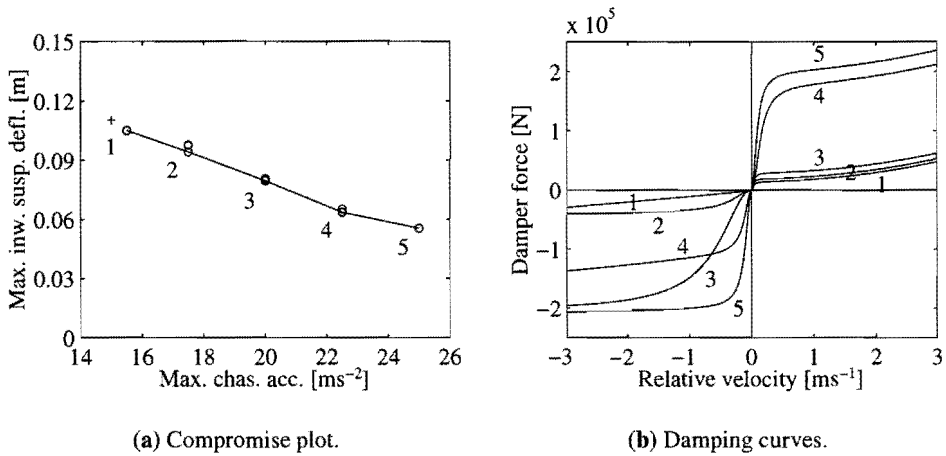


Figure 7.8: Optimum designs of the quarter car model with stroke dependent damping for several acceleration bounds. The plus sign marks the original axle suspension without extra damping.

It is quite difficult to preselect the maximum acceleration bound beforehand. To judge the suspension capability, the design engineer would probably prefer a compromise plot of the maximum attainable suspension deflection versus the maximum chassis acceleration. Optimizations have been carried out for a range of acceleration bounds varying from 15.5 ms^{-2} to 25 ms^{-2} . A design with a maximum acceleration lower than 15 ms^{-2} was not found. Accelerations did not exceed 25 ms^{-2} , whatever the curve of the stroke dependent damping. Furthermore, optimizations have been restarted from some different initial designs. The calculated optima are plotted in Figure 7.8(a). The solid line connects the designs with the smallest negative suspension deflection. Designs below this line can probably not be realized. Though, one hundred percent guarantee cannot be given since the global optima may not yet have been found.

Figure 7.8(a) shows that the additional stroke dependent damping can hardly reduce the maximum suspension deflection if the maximum acceleration is limited to the level of the original axle suspension. The original design without extra damper is marked with a plus sign and nearly lies on the compromise line. The stroke dependent damper can reduce the maximum negative suspension deflection, but only at the expense of increased maximum accelerations for the *incidental* road undulations. Remark that the comfort for (stochastic) road conditions with suspension deflections above -0.04 m is not affected.

Some trends in the optimum design variable values are present (see Figure 7.8).

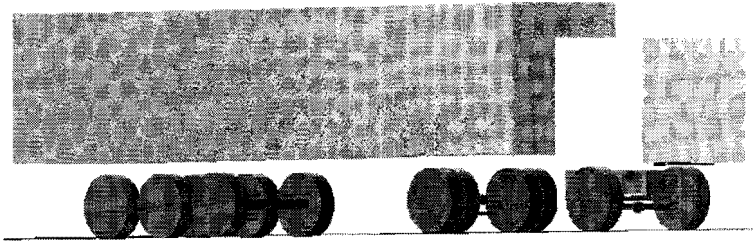


Figure 7.9: Impression of the full-scale DADS model of a DAF FT95 tractor-semitrailer combination.

All optimum damping curves tend towards a rather flat blow-off for the rebound stage, and, to a less extent, for the compression stage as well (small b_6 and b_3). Furthermore, the bleed on the rebound side (b_4) is usually near its upper bound value. The other optimum design variable values show a much larger variation. The bleed (b_1) and preload (b_2) of the compression stage tend to increase for higher maximum allowed acceleration. This corresponds with what one would expect to happen. They also seem slightly correlated: lowering b_1 can partly compensate the effect of enlarging b_2 , and vice versa. The same sort of correlation can be observed for b_4 and b_5 . This confirms the existence of multiple local optima. Finally, the design variable values of the compression stage (b_1 , b_2 , and b_3) are much more determined by the maximum allowed acceleration than the design variables of the rebound stage.

7.4 Full-Scale Model

7.4.1 Problem Description

The quarter car model presented in the previous section is the most simple model incorporating the compromise between suspension working space and comfort. It requires only few computer time, and numerical optimization can be carried out without great difficulty. However, due to its simplicity, the quarter car model is not highly accurate. A great deal of the dynamic behavior of the truck is not included. More complex models show a better correspondence with the actual behavior. Bekkers (1995), for example, developed a full-scale (3-D) DADS model of a DAF FT95 tractor-semitrailer combination (see Figure 7.9).

The 3-D DADS multibody model is a quite close reproduction of the truck tractor and semitrailer. The main model components of the tractor are chassis, front and rear axles, axle suspensions, tires, engine, steering system, cabin, and

cabin suspension. Even the fuel tank has been included. The semitrailer consists of a frame, three axles, and axle suspensions. Each component is modeled by one or more rigid bodies and couplings. Parameter values of nonlinear damper and spring characteristics are based on experimentally determined behavior. The complete model has 34 degrees of freedom, and is used in the optimization like the quarter car model in the previous section.

The full-scale model has been slightly adapted for optimization purpose. The main adjustment is the removal of bump stops. The bump stops represent rubber stops that restrict the suspension deflection working space. Contact with a bump is more or less an impact situation, leading to a series of peak accelerations added to the global response. These peak accelerations are of no interest for the design of the stroke dependent damper. The damper should reduce the suspension deflections to avoid the bump contacts. Besides, peak accelerations raise serious difficulties for the numerical optimization, and give rise to high computational cost of the numerical analysis.

Two road surfaces are considered: the traffic hump and the wave. Calculations for the quarter car model showed that the railway crossing does not significantly contribute. The truck follows a straight line such that both left and right wheels meet the road undulation at exactly the same time. So, left and right side of the truck behave identically, apart from small differences due to asymmetrical mass distribution (left and right are defined from the driver's point of view).

The suspension deflection and vertical chassis acceleration at the front left side of the truck are shown in Figure 7.10 for traffic hump and wave. During the first five seconds the truck drives on a flat road. At $t = 5$ s the front wheels meet the road undulation. The wave causes a maximum negative suspension deflection of about 0.12 m and a maximum acceleration of 20 ms^{-2} . The global behavior of relative displacements and accelerations shows a good resemblance with the responses calculated for the quarter car model (cf. Figure 7.5). The timing of maxima and minima corresponds reasonably well. However, the amplitudes show larger discrepancies. Differences of a factor two are present for some of the extrema. Such large differences are possible since the quarter car model does not include all vehicle dynamics.

The front axle suspensions on both sides of the truck model are extended with an extra nonlinear damping that has to supply the stroke dependent damping for negative suspension deflections below -0.04 m. The optimization problem is exactly the same as mentioned in Section 7.3, except for the constraints on the maximum tire deflection which are removed. So, only the suspension deflection and vertical chassis acceleration at the front left side of the truck are included in the optimization problem.

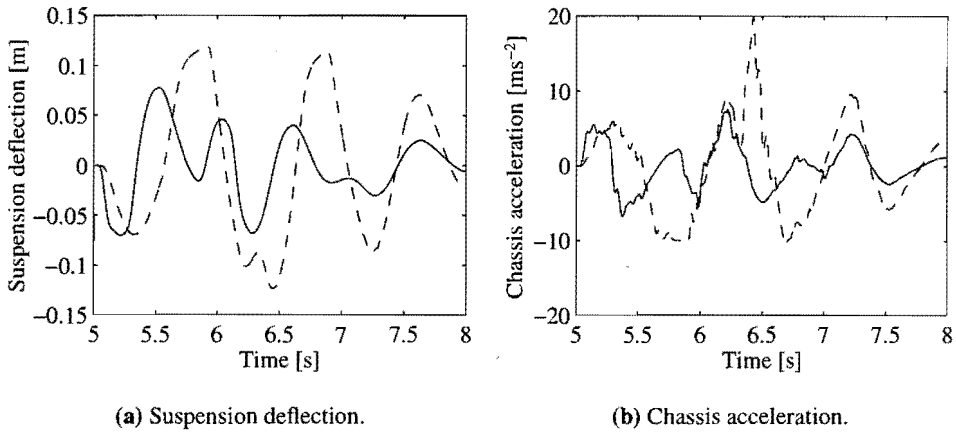


Figure 7.10: Displacement and acceleration responses of the full-scale model for the traffic hump (solid), and wave (dashed).

Figure 7.10(b) shows higher frequencies in the calculated accelerations compared with the quarter car model. Therefore, the density of the time point distribution is increased to 801 points on a smaller interval of 5 to 7 s. The integration accuracy is reduced to 10^{-4} , together with the finite difference step $\beta = 10^{-2}$, and the convergence parameters $\text{objacc} = 1\%$ and $\text{vioacc} = 1\%$. This is much cheaper in computational cost compared with an accuracy of 10^{-7} . Even then, a numerical analysis of the full-scale model is still two hundred times more expensive than a quarter car analysis. Therefore, the number of design variables is reduced as well. Design variables b_3 , b_4 and b_6 are kept fixed at $0.7 \cdot 10^3 \text{ Nsm}^{-1}$, $22 \cdot 10^6 \text{ Ns}^2\text{m}^{-2}$ and $0.7 \cdot 10^3 \text{ Nsm}^{-1}$, respectively. Optimizations in the previous section have shown that these design variables tend to go to their upper or lower bounds.

7.4.2 Optimization Results

For a maximum allowed chassis acceleration of 20 ms^{-2} the optimization history is shown in Figure 7.11. The initial design equals the optimum parameter values as found for the quarter car model in Table 7.3. Convergence occurred after fourteen cycles. The bleed during compression b_1 increases about a factor ten to $5.8 \cdot 10^6 \text{ Ns}^2\text{m}^{-2}$. Preload b_2 is halved towards: $10 \cdot 10^4 \text{ N}$. The rebound stiffness b_5 hardly changes, $33 \cdot 10^3 \text{ N}$ instead of $29 \cdot 10^3 \text{ N}$. So, the first two design variables show a significantly different value compared with the optimum design of the quarter car model. Furthermore, the wave is decisive instead of the traffic hump.

The suspension deflection and vertical chassis acceleration of the calculated

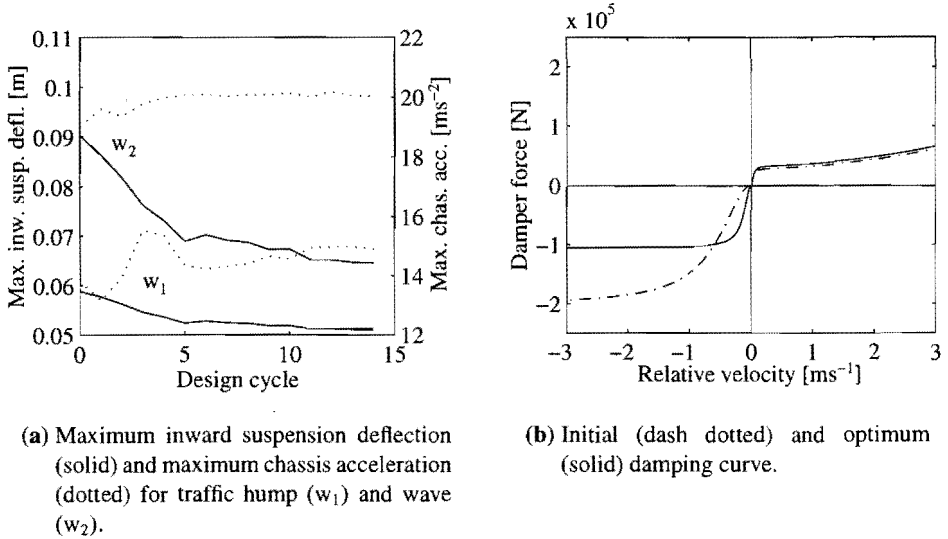


Figure 7.11: Optimization history of the full-scale model with stroke dependent damping for a maximum acceleration of 20 ms^{-2} .

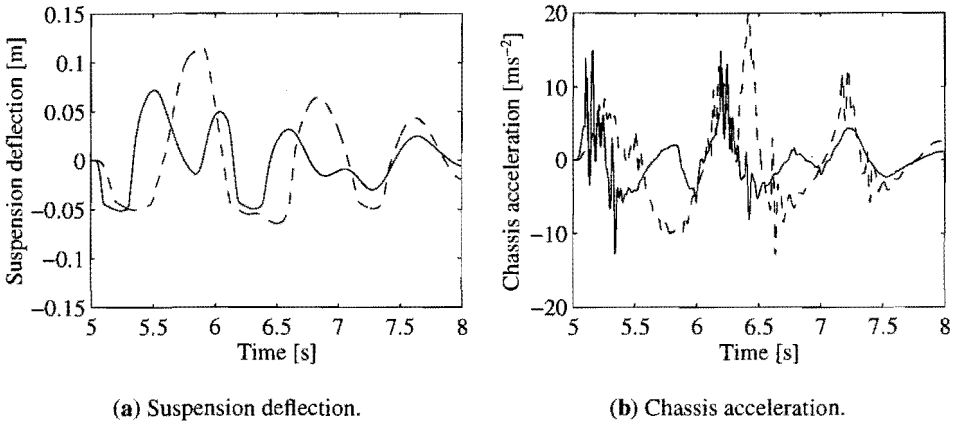


Figure 7.12: Displacement and acceleration responses of the full-scale model with optimized stroke dependent damping for the traffic hump (solid), and wave (dashed).

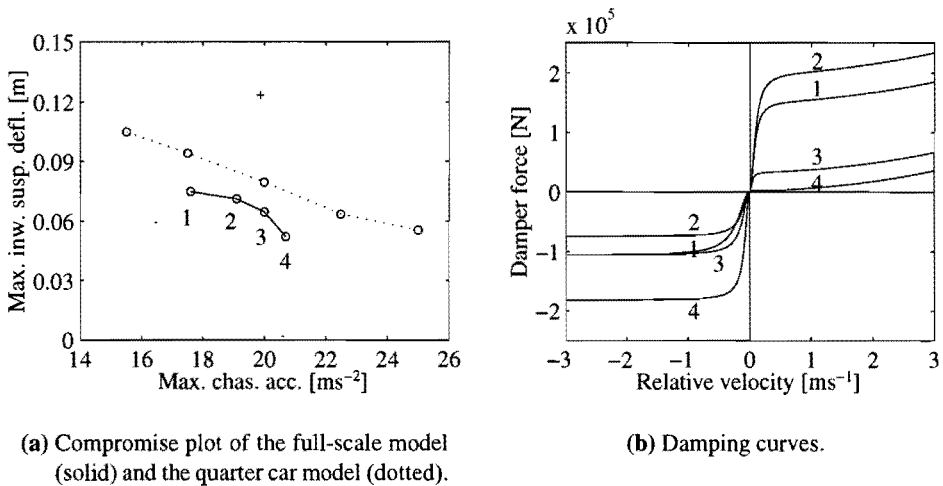


Figure 7.13: Optimum designs of the full-scale model with stroke dependent damping for several acceleration bounds. The plus sign marks the original axle suspension without extra damping.

optimum design are plotted in Figure 7.12. It is clearly visible that the negative suspension deflections below -0.04 m have been suppressed. For the traffic hump, this comes at the expense of an increase of the maximum acceleration. It is striking that this seems to be mainly caused by a more pronounced presence of the high frequent vibrations on the acceleration response. The global response stays at about the same magnitude. The same effect can be observed for the wave, but to a far less extent. Here the maximum acceleration is hardly affected by the amplification of the vibration.

Some additional optimizations have been carried out for other acceleration bounds than 20 ms^{-2} . Optimum designs with accelerations lower than 17.5 ms^{-2} , and higher than 21 ms^{-2} have not been found. Figure 7.13(a) shows the compromise plot, based on optimizations with acceleration bounds 17.5 , 19 , 20 , and 22.5 ms^{-2} . The optimization runs for the former two bounds did not converge. The desired accuracy could not be reached, which is probably caused by the combination of inaccurate sensitivities and high frequencies in the acceleration responses. For each of these two bounds, the plotted optimum design corresponds with the design nearest to the constraint bound.

The compromise plot now clearly lies below the original full-scale model design without stroke dependent damping (plus sign). So, in contrary to the quarter car model, the maximum negative suspension deflection can be reduced without

increasing the maximum acceleration. However, it should be noted that this is mainly caused by the fact that the maximum acceleration due to the wave is hardly affected by the damping curve. Similar to the quarter car problem, the optimum design variable values of b_1 , b_2 and b_5 are influenced by the acceleration bound. Figure 7.13(b) shows that the compression bleed b_1 increases if a higher maximum acceleration is allowed. The preload b_2 tends to increase as well. Parameter b_5 shows a completely different behavior compared with the quarter car model (compare Figure 7.13(b) and Figure 7.8(b)). The rebound preload decreases instead of increases as a function of the maximum acceleration.

7.5 Conclusion and Discussion

Inclusion of stroke dependent damping in the front axle suspension of a truck can reduce large (inward) deflections appearing at incidental road disturbances. In this way the increase of suspension deflections due to stiffness reduction can be compensated. At the same time, it may lead to increased chassis accelerations the moment the extra damping comes into play. This possible loss of comfort is preferable above contact with bump stops that restrict the suspension working space. The most important impediment, however, is the required stroke dependent damping force. In the current study, the damper force may rise to 200 [kN], which is about two and a half times as high as the damper force of the original absorber. The question is whether this is an cost-effective solution. Possibly, the optimization problem definition should be reconsidered, e.g. by decreasing the design space or by including the damper force as objective function or constraint.

No single optimum damping curve has been found that generally yields the best stroke dependent damping. The shape of the curve depends upon the maximum allowed acceleration. Still, the optimization results point towards the following design rules for the stroke dependent damping. First of all, the blow-off stiffness can remain small. Furthermore, the bleed and blow-off preload are the most important variables influencing the compromise between suspension deflection and chassis acceleration. To reduce inward suspension deflections, the compression side of the damping curve is of main interest. Choose the compression bleed and preload as high as possible until the selected acceleration bound is reached. Most attention should be paid to the preload since this parameter defines the magnitude of the damping force that can be generated. One may be tempted to apply the high bleed and preload levels to the rebound side as well. However, high rebound damping can cause the axle suspension to show a bad recovery and to stay at large negative deflections. This means that bump contacts may occur for a second road disturbance

or steer maneuver shortly after the first one.

The design optimization was first applied to the quarter car model before the full-scale model was considered. The basic idea is that a rather simple model gives the opportunity to do several optimizations runs and to get a feeling for the design problem. If necessary, the optimization problem can be reformulated or the mechanical model can be adapted. Hopefully, trends become visible, and the designer starts to form an idea of what is actually searched for. The latter is often far from a foregone conclusion. Then the step can be made towards a more complex analysis model. Starting point is that the basic response behavior corresponds for both models. The design engineer should now be able to do the optimization of the complex model in a structured way without too much experimentation.

For a well phased optimization the step from quarter car model to full-scale model has been pretty large. A reasonable resemblance of the global response behavior is present. Though, the discrepancies cause the compromise lines to deviate. Large differences are especially present for the chassis acceleration. A more detailed confrontation of quarter car and full-scale model is desired to find out the precise cause of the deviation, and to determine which parts of the tractor-semitrailer behavior can in which parts cannot be neglected for the design optimization. Possibly, the quarter car model should be replaced by, for example, a six-DOF vehicle model. If the model can be kept simple the number of road excitation inputs in the optimization problem can be increased without difficulty.

Anyway, the design optimization tool has proven its value. For the quarter car model reasonable to good convergence was found. The moment the extra damping comes in, a sharp acceleration peak may be present, especially for the traffic hump. This suggests to let the integration procedure define which time points are included in the optimization, instead of an equal distribution on the time interval. However, since an external multibody software package was used, equally distributed time points were the only option that could be realized. Difficulties observed for the full-scale model are mainly due to inaccurate finite difference sensitivities. This underlines the necessity of an integrated multibody analysis and design sensitivity analysis. If, from a practical point of view, sensitivities can only be obtained by finite differences with a poor accuracy, a multi-point approach may be considered instead.

Chapter 8

Conclusions and Recommendations

8.1 Problem Description

Well-established software is available for multibody system analysis as well as for numerical optimization. The ultimate goal is to integrate multibody analysis and optimization into one general purpose design tool. This requires the analysis and optimization modules to work for a variety of multibody design problems, ranging from mechanism kinematics to vehicle system dynamics. The resulting optimization problems may combine nonlinear, time dependent or non-smooth objective function and constraints with a computationally expensive numerical multibody analysis.

The design engineer plays an important role in the optimization process. He is responsible for the mathematical formulation of the optimization problem. This should be carefully considered since it largely determines whether a satisfactory optimum design is found. A serious difficulty is that the designer often does not precisely know how to mathematically formulate the multibody design problem beforehand. It is very likely that during the optimization he wants to remove or change objective function, constraints and design variables. Therefore, graphical means have to be available, not only for modeling, but for optimization purpose as well. Haug *et al.* (1986) and Erdman (1995) stress the importance of an *interactive* computer aided design tool for a successful design optimization.

Both the computational complexity as well as the demand for user-interaction hinders the success of a direct coupling between multibody analysis and optimization algorithm. The same sort of problems occurs for finite element structural anal-

ysis. In this field of application, the incorporation of approximation techniques in the optimization has proven to be an effective approach. However, within the context of multibody systems these techniques are hardly used. The objective of this thesis is to investigate whether approximation concepts are valuable for the design optimization of multibody systems.

8.2 How Approximation Concepts can help

Approximation concepts quite naturally provide an interface in between numerical multibody analysis and optimization. Within a certain search subregion, an approximate optimization problem is built based on multibody calculations for a single or multiple design points. The approximate optimization problem can be solved by the optimizer without any call to the analysis routines, which is computationally advantageous. Usually a sequence of approximate optimization cycles follows to arrive at the final optimum design. For each new cycle the optimization problem can be reformulated. Furthermore, the approximations can be improved, for example by introduction of intermediate design variables and intermediate response variables that represent explicit or physical nonlinear behavior.

The multibody responses calculated from the equations of motion are often time dependent. They can be included in the approximate optimization problem by means of time discretization. As a result, a large number of time point constraints may arise, which is disadvantageous for the optimization cost. Therefore, several researchers remove time by equivalent integral constraint formulations, despite potential numerical difficulties. However, constraint deletion can greatly reduce the number of time point constraints in the approximate optimization problem. Only the most important constraints that are near or violate their constraint bound need to be retained. In this way, it is usually possible to obtain an approximate optimization problem of manageable proportions, even for complex multibody systems with many time points and responses.

Two classes of approximation concepts have been identified that are especially suitable for the design optimization of multibody systems. Multibody optimum design problems may have ten or more design variables and a large number of constraints. Local and mid-range approximation concepts can handle this type of problems. For smooth objective function and constraints single-point local approximations are most suited. Numerical methods are available to efficiently and accurately compute the required design sensitivities with respect to the design variables. Whenever noisy functional behavior occurs, a multi-point mid-range concept that does not use gradient information is appropriate. In that case for each cycle a small

experimental design of multibody analyses is carried out to build the approximations. Both the local and mid-range concepts allow a constraint deletion and the introduction of intermediate design variables and intermediate response quantities.

8.3 Approximate Optimization Tool

Multibody design optimization software has been developed based on the local and multi-point mid-range concepts. The software uses time discretization and constraint deletion to generate the approximate optimization problems. Basically, it starts from linear approximations of objective function and constraints. However, intermediate design variables and intermediate responses can be defined to improve the single-point linear approximations. The multi-point approximations can benefit from intermediate variables and responses as well, but this option has not yet been implemented. Finally, a move limit strategy is present to adapt the size of the search subregion during the sequential approximate optimization process according to the approximation errors and the convergence behavior.

Several multibody optimum design problems have been successfully solved by means of the developed optimization tools. Two analytical examples were studied first: a single degree of freedom impact-absorber and a slider-crank mechanism. They illustrated the beneficial effect of intermediate response quantities, and the ability of the multi-point approach to solve non-smooth functional behavior provided that a reasonable estimation of the magnitude of the noise is given. Furthermore, the impact-absorber example showed that oscillations may occur in the convergence stage of the optimization if the constraint deletion only retains a single time point constraint for each local maximum. Multiple time point resolved this problem, which comes, however, at the expense of an increased number of time point constraints in the approximate optimization problem. An approximation concept with adjustable conservativeness may be able to include only one time point constraint for each maximum, and still avoid oscillations.

Three larger design applications were studied. To this end, the optimization software was coupled with the commercial multibody analysis packages MECANO (Samtech, 1994), MADYMO (TNO, 1994) and DADS (CADSi, 1995). The first application concerned the stress constrained design of a four-bar mechanism using local approximations. It proved that the selected intermediate design variables and responses yielded better function approximations, but that vibrations affected the quality of the approximations adversely. In the second application, multi-point approximations were used to optimize a vehicle restraint system. The safety criteria appeared to have a non-smooth behavior as function of the design variables.

Finally, a stroke dependent damper for the front axle suspension of a truck was optimized. Various optimization runs were carried out for a two degree of freedom and a full-scale truck model using single-point linear approximations. This application clearly showed that the final optimum design heavily relies on the choices made by the designer.

8.4 Intermediate Variables and Responses

High quality objective function and constraint approximations yield a fast and robust optimization process. The search subregions can be large and there is no need for (repeated) move limit reduction. In the optimization tool the basic approximation is linear. Therefore, it is important to search for intermediate design variables and intermediate response quantities that yield the best linear approximations. Often explicit or easily computable relations are present between intermediate responses and objective function and constraints on the one hand, and between design variables and intermediate design variables on the other. As a result, any nonlinearity present in these relationships will be included in the approximate optimization problem and improve the approximations.

For multibody systems several intermediate design variables and intermediate response quantities can be identified. Linkages design, for example, often aims at the realization of some desired path of motion. The corresponding objective function may have a nonlinear functional relation, such as the sum of squared distances between the desired and generated positions for the slider-crank mechanism example. Also many comfort criteria incorporate explicit nonlinear behavior, as illustrated by the impact-absorber. For both slider-crank mechanism and impact-absorber a significant reduction of the number of multibody analyses during optimization was found compared to a direct coupling with an SQP-algorithm. The stress constrained four-bar mechanism showed that for beam elements it is convenient to use cross sectional areas and bending moments as intermediate design variables and intermediate responses. However, vibrations induced by the motion of the system spoiled their beneficial effect. Possibly, intermediate variables and responses may be found that predict the change of the vibration frequencies.

8.5 Design Sensitivity Analysis

Design sensitivities can be obtained in three different ways. Finite differencing is probably the most simple method, but it is computationally expensive and suffers from numerical inaccuracies. A better approach is to analytically differentiate the

governing equations of motion with respect to the design variables. Sensitivities can then be calculated by solving the sensitivity equations for each design variable. This is called the direct differentiation approach. In cases where sensitivities are required just for a few (time point) responses, it may be computationally more efficient to first rewrite the sensitivity equations by means of adjoint variables. This adjoint variable method requires the solution of sensitivity equations for each response instead of each design variable.

The governing equations of multibody systems are either algebraic, ordinary differential, or differential-algebraic equations. The same type of sensitivity equations follow using the direct or the adjoint method. However, the adjoint differential and differential-algebraic sensitivity equations have to be integrated backward in time, instead of forward. As a consequence, during forward integration of the equations of motion all responses required for the sensitivity calculations have to be stored, and interpolated during backward integration. Therefore, the adjoint method is much more difficult to implement compared with direct differentiation, and requires a lot of input and output operations as well as data storage.

For differential and differential-algebraic equations, only the adjoint method can truly benefit from the constraint deletion. Adjoint sensitivity equations can be assembled for each time point response that is included in the approximate optimization problem. Then it is important that the constraint deletion retains as few constraints as possible. Multiple time points at local response maxima will usually spoil the computational advantage of the adjoint method in comparison with the direct method. So the adjoint strategy really needs a constraint screening that includes only the most critical maximum time points, and an approximation concept that can overcome potential oscillations.

Intermediate variables and responses also point towards direct differentiation as the most appropriate sensitivity method if approximate optimization with time discretization is applied. The direct method calculates the design sensitivities along with positions, velocities, accelerations and Lagrange multipliers of the equations of motion. Objective function and constraints are related to these multibody responses. So there is a clear distinction between, on the one hand, the multibody responses arising from the equations of motion, and, on the other hand, the objective function and constraints present in the optimization problem formulation. Any nonlinearity present in their mutual relations can be defined as an intermediate response quantity. On the contrary, the adjoint variable approach starts from a summation or integral function of multibody responses. This relation is used to avoid the explicit calculation of the sensitivities for all responses and time points. However, it may as well hinder the utilization of intermediate design variables and intermediate response quantities.

8.6 Further Recommendations

Approximation concepts can be effectively applied for the design optimization of multibody systems. They will prove to be a valuable part in a general purpose multibody design tool. Other important elements are the multibody analysis and sensitivity analysis, the graphical user-interface, a database combined with for example an expert system, and a control program to manage the complete design process. An important future step is to put all parts together and build a completely integrated and interactive design tool. This will put a different complexion on each of the individual elements.

The objective to develop a general purpose multibody design tool raises several new questions. For example, is symbolic computation the only practical way to assemble the sensitivity equations of motion, or can numerically oriented multibody packages be extended with a design sensitivity analysis as well without an immense burden of programming work? Can a design sensitivity analysis module be developed that efficiently and accurately computes sensitivities for a broad range of multibody systems? Is the optimization module able to deal with the great variety of multibody optimum design problems, or is it necessary to distinguish branches of applications and develop separate optimization strategies? And, how should the graphical interface look alike for a fast and robust optimization?

Other research topics can be identified as well. As an example, the basic but challenging problem to define constraints on lock-up and bifurcation is mentioned. The difficulty is that there are actually just two alternatives: a mechanism locks or doesn't, with no alternative in between. The moment of transition cannot be easily predicted by a simple mathematical function. Another interesting research area is topology optimization of multibody systems. This thesis primarily concentrated on dimensional optimization. The optimization problem becomes far more complex if the number as well as the placement of bodies and couplings may vary.

Appendix A

Move Limit Strategy

The choice of the move limits has a great influence on the behavior of the sequential approximate optimization process. Especially if the approximations are less accurate, the move limit strategy usually is of vital importance. Only approximation concepts like convex linearization (Fleury and Braibant, 1986) and the method of moving asymptotes (Svanberg, 1987, 1996) do not need move limits due to their conservative nature. In this appendix the move limit strategy is described that has been implemented in the developed optimization software.

The move limit strategy determines new search subregions on the basis of maximum approximation errors, maximum constraint violations and convergence behavior of the design variables. It is executed after an approximate optimization cycle has been completed, including the analysis of the approximate optimum solution. Then, it adjusts the move limit factors of Equations (3.22) and (3.27) for the next approximate optimization cycle. In the sequel, it is assumed that design cycle q has been finished, and that the corresponding approximation errors and constraint violations have been calculated. Objective function and design variable values of the previous cycle are used as well.

The move limit strategy is presented by a program text containing conditionally execute statements (syntax: `if-elseif-else-endif`) and repeat statements (`for-endfor`). In this text some abbreviations are used. The objective function approximation error, the maximum constraint approximation error, and the overall maximum approximation error are denoted by E_f , E_g , and E , respectively. The maximum constraint violation is written as V . They follow from Equations (3.15), (3.16) and (3.18). The objective function values of the cycle start design $F(b_0^{(q)})$ and the approximate optimum solution $F(b_*^{(q)})$ are represented by F_0 and F_{opt} , respectively. The percentage of change of the objective function value during the

current cycle is given by:

$$Fchg = \left| \frac{F(\mathbf{b}_*^{(q)}) - F(\mathbf{b}_0^{(q)})}{F(\mathbf{b}_0^{(q)})} \right| \times 100\%. \quad (\text{A.1})$$

Furthermore, a boolean is used to identify whether or not a direction change of the objective function value has occurred compared with the previous approximate optimization cycle:

$$Fdrchg = \begin{cases} \text{true} & \text{if } \left(F(\mathbf{b}_*^{(q)}) - F(\mathbf{b}_0^{(q)}) \right) \left(F(\mathbf{b}_0^{(q)}) - F(\mathbf{b}_0^{(q-1)}) \right) < 0, \\ \text{false} & \text{if } \left(F(\mathbf{b}_*^{(q)}) - F(\mathbf{b}_0^{(q)}) \right) \left(F(\mathbf{b}_0^{(q)}) - F(\mathbf{b}_0^{(q-1)}) \right) \geq 0. \end{cases} \quad (\text{A.2})$$

The same type of booleans can be defined for the design variable changes:

$$bdrchg(i) = \begin{cases} \text{true} & \text{if } (b_{*i}^{(q)} - b_{0i}^{(q)})(b_{0i}^{(q)} - b_{0i}^{(q-1)}) < 0, \\ \text{false} & \text{if } (b_{*i}^{(q)} - b_{0i}^{(q)})(b_{0i}^{(q)} - b_{0i}^{(q-1)}) \geq 0. \end{cases} \quad (\text{A.3})$$

A true value of $bdrchg(i)$ means that the corresponding design variable value has changed in opposite direction compared with the previous design cycle. This may indicate the start of oscillations. Using these abbreviations the move limit strategy is described below.

First of all, the approximation errors and the maximum constraint violation are checked. The approximate optimum design of the current design cycle is not accepted if the maximum approximation error is too large, or too high an infeasibility occurs starting from a feasible or nearly feasible solution. The latter precondition is met by an approximate optimization problem without constraint relaxation (see Section 3.4). In the case of rejection, the current design cycle is repeated with reduced move limits (usually $\beta_1 = 0.5$ has been selected). These statements are represented by the following program text:

```
if (E > errmax) or ((V > viomax) and (not relax)) then
  accept := false
  for i := 1,n
    m(i) := beta1 * m(i)
  endfor
```

After the proposed optimum solution has been accepted, the convergence criteria are checked. The objective function value, the constraint violation, as well as the

objective function and constraint approximation errors should have the requested accuracy, as given by `objacc` and `vioacc`, respectively. The corresponding program text is:

```

else
  accept := true
  if (Fchg < objacc) and (V < vioacc) and
    (Ef < objacc) and (Eg < vioacc) then
    convrg := true

```

If no convergence has occurred, the move limit factors of the next approximate optimization cycle are determined, for each design variable separately. Starting point is to halve the move limits ($\beta_3 = 0.5$) of a design variable if it starts to oscillate together with an oscillating or highly accurate objective function value ($\beta_2 = 4$). If this is not the case, but the objective function increases, a less rigorous move limit reduction is applied ($\beta_4 = 0.75$). However, this may only happen if no constraint relaxation is present.

```

else
  convrg := false
  for i := 1,n
    if (bdrchg(i)) and
      ((Fdrchg) or (Fchg < beta2 * objacc)) and
      (not relax) then
      m(i) := beta3 * m(i)
    elseif (Fopt > F0) and (not relax) then
      m(i) := beta4 * m(i)

```

In all other cases the optimization process is not assumed to be near the final optimum solution. The move limits are enlarged or decreased just to keep a steady course of the optimization. Whenever the maximum approximation error or the maximum constraint violation tends to become too large, move limits are reduced. On the other hand, very small values may indicate that a much faster optimization process can be obtained by increasing the move limits. Two cases are distinguished: 1) the design variable of the approximate optimum solution of the current cycle has been bounded by the corresponding move limit, or 2) the move limit is not active at all. In the former case, move limits can be either increased ($\beta_5 = 4/3$) or decreased ($\beta_4 = 3/4$), while in the latter case only move limit reduction can occur, since an increase is expected to have no effect.

```

else
  if (mvlact(i)) and (not bndact(i)) then
    if (E < errsml) and (not E = 0) and
      ((V < viosml) or (relax)) and
      (not bdrchg(i)) then
      m(i) := beta5 * m(i)
    elseif (E > errlrg) or
      ((V > violrg) and (not relax)) then
      m(i) := beta4 * m(i)
    end
  else
    if (E > errlrg) or
      ((V > violrg) and (not relax)) then
      m(i) := beta4 * m(i)
    endif
  endif
endif
endif
endif
endif

```

Booleans `mvlact(i)` and `bndact(i)` are true when the move limits and the design variable bounds are active, respectively. Move limit enlargement is not allowed if no approximations of objective function or constraint are included in the approximate optimization problem (represented by `not E = 0`). This may, for example, occur if none of the constraints is active or potentially active, because the cycle start design is far away from the constraint boundaries.

Eight parameters have to be set before the start of the optimization: `objacc`, `vioacc`, `errsml`, `errlrg`, `errmax`, `viosml`, `violrg`, and `viomax`. They are summarized on page xvii. The user has to select these parameters according to his experience and the accuracies of the responses that are obtained from the numerical analysis. Finally, it is remarked that this algorithm will not converge if no feasible solution is present at all: the condition `maxg < viomax` will never be satisfied. For convergence, one may replace the maximum constraint violation `maxg` by the maximum constraint violation change in the convergence criterion.

Bibliography

Abramowitz, M. and Stegun, I. A., editors (1965). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, Inc., New York.

Adelman, H. M. and Haftka, R. T. (1986). Sensitivity analysis of discrete structural systems. *AIAA Journal*, **24**, 823–832.

Adriaens, J. M. T. A. (1995). *Crashworthiness Design Optimisation using Multipoint Approximations*. Master's thesis, Eindhoven University of Technology, Eindhoven. WFW 95.180, confidential.

Afimawala, K. A. and Mayne, R. W. (1974). Optimum design of an impact absorber. *ASME Journal of Engineering for Industry*, **96**, 124–130.

Ashrafioun, H. and Mani, N. K. (1990). Analysis and optimal design of spatial mechanical systems. *ASME Journal of Mechanical Design*, **112**, 200–207.

Barthelemy, J.-F. M. and Haftka, R. T. (1993). Approximation concepts for optimum structural design - a review. *Structural Optimization*, **5**, 129–144.

Barthelemy, J.-F. M. and Hall, L. E. (1995). Automatic differentiation as a tool in engineering design. *Structural Optimization*, **9**, 76–82.

Bathe, K.-J. (1996). *Finite Element Procedures*. Prentice-Hall, Inc., Englewood Cliffs.

Bekkers, F. P. J. (1995). *Modular based Vehicle Modelling*. Master's thesis, Eindhoven University of Technology, Eindhoven. WFW 95.078.

Bennett, J. A. and Park, G. J. (1995). Automotive occupant dynamics optimization. *Shock and Vibration*, **2**, 471–479.

- Bestle, D. (1994). *Analyse und Optimierung von Mehrkörpersystemen*. Springer-Verlag, Berlin.
- Bestle, D. and Eberhard, P. (1992). Analyzing and optimizing multibody systems. *Mechanics of Structures and Machines*, **20**, 67–92.
- Bestle, D. and Seybold, J. (1992). Sensitivity analysis of constrained multibody systems. *Archive of Applied Mechanics*, **62**, 181–190.
- Bischof, C. H. (1996). On the automatic differentiation of computer programs and an application to multibody systems. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 41–48, Dordrecht. Kluwer Academic Publishers.
- Bosio, A. C. and Lupker, H. A. (1991). Design of experiments in occupant simulation. *SAE Technical paper*. 910891.
- CADSi (1995). *DADS Reference Manual, version 8.0*. Computer Aided Design Software, Inc., Coralville.
- Chang, C. O. and Nikravesh, P. E. (1985). Optimal design of mechanical systems with constraint violation stabilization method. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **107**, 493–498.
- Demić, M. (1989). Optimisation of the characteristics of the elasto-damping elements of a passenger car by means of a modified Nelder-Mead method. *International Journal of Vehicle Design*, **10**, 136–152.
- Dias, J. P. and Pereira, M. S. (1994). Design for vehicle crashworthiness using multibody dynamics. *International Journal of Vehicle Design*, **15**, 563–577.
- Eberhard, P. (1996). *Zur Mehrkriterienoptimierung von Mehrkörpersystemen*. Number 11 (227) in Fortschritt-Berichte VDI. VDI Verlag, Düsseldorf.
- Eberhard, P., Bestle, D., and Piram, U. (1996). Optimization of damping characteristics in nonlinear dynamic systems. In N. Olhoff and G. I. N. Rozvany, editors, *WCMSO-1 First World Congress of Structural and Multidisciplinary Optimization*, pages 863–870. Pergamon.
- Elbeheiry, E. M., Karnopp, D. C., Elaraby, M. E., and Abdelraaouf, A. M. (1995). Advanced ground vehicle suspension systems - a classified bibliography. *Vehicle System Dynamics*, **24**, 231–258.

Erdman, A. G. (1995). Computer-aided mechanism design: now and the future. *ASME Special 50th Anniversary Design Issue*, **117** (B), 93–100.

Etman, L. F. P. (1994). Design and analysis of computer experiments: the method of Sacks et al. Technical Report WFW 94.098, Eindhoven University of Technology, Eindhoven.

Etman, L. F. P., Adriaens, J. M. T. A., van Slagmaat, M. T. P., and Schoofs, A. J. G. (1996a). Crash worthiness design optimization using multipoint sequential linear programming. *Structural Optimization*, **12**, 222–228.

Etman, L. F. P., van Campen, D. H., and Schoofs, A. J. G. (1996b). Optimization of multibody systems using approximation concepts. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 81–88, Dordrecht. Kluwer Academic Publishers.

Etman, L. F. P., Thijssen, E. J. R. W., Schoofs, A. J. G., and van Campen, D. H. (1994). Optimization of multibody systems using sequential linear programming. In B. J. Gilmore, D. A. Hoeltzel, D. Dutta, and H. A. Eschenauer, editors, *ASME Advances in Design Automation*, volume DE 69-2, pages 525–530.

Fleury, C. and Braibant, V. (1986). Structural optimization: a new dual method using mixed variables. *International Journal for Numerical Methods in Engineering*, **23**, 409–428.

Fleury, R. (1996). Analyse de sensibilité et optimisation de mécanismes. Mémoire Université de Liège.

Free, J. W., Parkinson, A. R., Bryce, G. R., and Balling, R. J. (1987). Approximation of computationally expensive and noisy functions for constrained nonlinear optimization. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **109**, 528–532.

Gabriele, G. A. (1993). Optimization in mechanisms. In A. G. Erdman, editor, *Modern Kinematics: Developments in the last forty years*, chapter 11. John Wiley & Sons, Inc., New York.

Gérardin, M. and Rixen, D. (1994). *Mechanical Vibrations: Theory and Application to Structural Dynamics*. John Wiley & Sons, Inc., Chichester.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization*. Academic Press, Inc., London, third edition.

- Goulou, C., Maillard, V., and Pernin, J. P. (1996). A coupled approach of simulation and optimization to design safety systems. In *15th Enhanced Safety of Vehicles - ESV Conference*. Paper No 96-S3-W-27.
- Grandhi, R. V., Haftka, R. T., and Watson, L. T. (1986). Design-oriented identification of critical times in transient response. *AIAA Journal*, **24**, 649–656.
- Haftka, R. T. and Adelman, H. M. (1989). Recent developments in structural sensitivity analysis. *Structural Optimization*, **1**, 137–151.
- Haftka, R. T. and Gürdal, Z. (1992). *Elements of Structural Optimization*. Kluwer Academic Publishers, Dordrecht, third edition.
- Hansen, J. M. and Tortorelli, D. A. (1996). An efficient method for synthesis of mechanisms using an optimization method. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 129–138, Dordrecht. Kluwer Academic Publishers.
- Hardy, W. N., Khalil, T. B., and King, A. I. (1994). Literature review of head injury biomechanics. *International Journal of Impact Engineering*, **15**, 561–586.
- Haug, E. J. (1987). Design sensitivity analysis of dynamic systems. In C. A. Mota Soares, editor, *Computer Aided Optimal Design: Structural and Mechanical Systems*, volume F27 of *NATO ASI*, pages 705–755, Berlin. Springer-Verlag.
- Haug, E. J. (1989). *Computer Aided Kinematics and Dynamics of Mechanical Systems*, volume 1. Allyn and Bacon, Massachusetts.
- Haug, E. J. and Arora, J. S. (1979). *Applied Optimal Design: Mechanical and Structural Systems*. John Wiley & Sons, Inc., New York.
- Haug, E. J., Choi, K. K., and Komkov, V. (1986). *Design Sensitivity Analysis of Structural Systems*. Academic Press, Inc., London.
- Haug, E. J., Wehage, R. A., and Mani, N. K. (1984). Design sensitivity analysis of large-scale constrained dynamic mechanical systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **106**, 156–162.
- Hsieh, C. C. and Arora, J. S. (1984). Design sensitivity analysis and optimization of dynamic response. *Computer Methods in Applied Mechanics and Engineering*, **43**, 195–219.
- Lust, R. (1992). Structural optimization with crashworthiness constraints. *Structural Optimization*, **4**, 85–89.

- MathWorks (1996). *MATLAB Reference Guide, version 4*. The MathWorks, Inc., Natick.
- NAG (1991). *Fortran Library Manual, Mark 15*. Numerical Algorithms Group.
- Pagalday, J. M., Aranburu, I., Avello, A., and de Jalón, J. G. (1996). Multi-body dynamics optimization by direct differentiation methods using object oriented programming. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 213–220, Dordrecht. Kluwer Academic Publishers.
- Paradis, M. J. and Willmert, K. D. (1983). Optimal mechanism design using the Gauss constrained method. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **105**, 187–196.
- Pesch, V. J., Hinkle, C. L., and Tortorelli, D. A. (1996). Optimization of planar mechanism kinematics with symbolic computation. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 221–230, Dordrecht. Kluwer Academic Publishers.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, Cambridge, second edition.
- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, **4**, 409–435.
- Samtech (1994). *SAMCEF MECANO Manual, version 5.1*. Samtech, Liège.
- Schiehlen, W., editor (1990). *Multibody Systems Handbook*. Springer-Verlag, Berlin.
- Schoofs, A. J. G. (1987). *Experimental Design and Structural Optimization*. Ph.D. thesis, Eindhoven University of Technology, Eindhoven.
- Schoofs, A. J. G., Klink, M. B. M., and van Campen, D. H. (1992). Approximation of structural optimization problems by means of designed numerical experiments. *Structural Optimization*, **4**, 206–212.
- Sepulveda, A. E. and Schmit, L. A. (1993). Approximation-based global optimization strategy for structural synthesis. *AIAA Journal*, **31**, 180–88.
- Sohoni, V. N. and Haug, E. J. (1982). A state space technique for optimal design of mechanisms. *ASME Journal of Mechanical Design*, **104**, 792–798.

Spentzas, C. N. (1993). Optimization of vehicle ride characteristics by means of Box's method. *International Journal of Vehicle Design*, **14**, 539–551.

Svanberg, K. (1987). The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, **24**, 359–373.

Svanberg, K. (1996). A globally convergent version of MMA without linesearch. In N. Olhoff and G. I. N. Rozvany, editors, *WCMSO-1 First World Congress of Structural and Multidisciplinary Optimization*, pages 9–16. Pergamon.

Tak, T. and Kim, S. S. (1989). Design sensitivity analysis of multibody dynamic systems for parallel processing. In B. Ravani, editor, *ASME Advances in Design Automation*, volume DE 19-3, pages 9–16.

Tamis, P. (1994). *Optimisation of a Stroke Dependent Damper for the Rear Suspension of a Heavy Truck*. Master's thesis, Eindhoven University of Technology, Eindhoven. WOC/VT/R/94.55.

TNO (1994). *MADYMO User's Manual, version 5.1*. TNO Road-Vehicles Research Institute, Delft.

Toropov, V., van Keulen, F., Markine, V., and de Boer, H. (1996). Refinements in the multi-point approximation method to reduce the effects of noisy structural responses. In *Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume 2, pages 941–951.

Toropov, V. V., Filatov, A. A., and Polynkin, A. A. (1993). Multiparameter structural optimization using FEM and multipoint explicit approximations. *Structural Optimization*, **6**, 7–14.

Vanderplaats, G. N. (1979). Approximation concepts for numerical airfoil optimization. Technical Paper 1370, NASA.

Vanderplaats, G. N. (1984). *Numerical Optimization Techniques for Engineering Design*. McGraw-Hill, Inc., New York.

Vanderplaats, G. N. (1993). Thirty years of modern structural optimization. *Advances in Engineering Software*, **16**, 81–88.

Vermeulen, R. C. N. (1996). *Optimization of a Nonlinear Damper of a DAF FT95*. Master's thesis, Eindhoven University of Technology, Eindhoven. WFW 96.117, confidential (in Dutch).

Wimmer, J. and Rauh, J. (1996). Multicriteria optimization as a tool in the vehicle's design process. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 333–340, Dordrecht. Kluwer Academic Publishers.

Samenvatting

Veel werktuigkundige constructies kunnen gemodelleerd worden als een multibody-systeem dat bestaat uit een eindig aantal starre of flexibele lichamen. Bekende voorbeelden zijn voertuigen, mechanismen en robots. In de loop der jaren zijn computerprogramma's ontwikkeld die automatisch de bijbehorende bewegingsvergelijkingen kunnen opstellen en oplossen, en tevens zorg dragen voor een goede grafische visualisatie van de rekenresultaten. Dit soort programmatuur maakt het mogelijk om reeds vroegtijdig in het ontwerpproces het werktuigkundig ontwerp te analyseren en te verbeteren. Onwerpparameters, zoals massa's en veerstijfheden, kunnen eenvoudig worden gewijzigd om op die manier de gestelde ontwerpdoelen te realiseren met betrekking tot het kinematisch en dynamisch gedrag.

Numerieke optimaliseringsmethoden stellen de werktuigkundig ontwerper in staat om ontwerpverbeteringen op een systematische manier door te voeren. Bij meer dan twee ontwerpvariabelen verliest een ontwerper al gauw het overzicht, en wordt veelal ad hoc naar verbeteringen gezocht. Numerieke optimaliseringsalgoritmen kunnen daarentegen goed optimaliseringsproblemen met een groter aantal ontwerpvariabelen oplossen. Het is daarom wenselijk de programmatuur voor multibody-analyse zodanig uit te breiden dat ontwerpvariabelen, doelfunctie en beperkingen gedefinieerd kunnen worden, en dat een systematische oplossing van dit optimaliseringsprobleem mogelijk wordt.

Tot dusver zijn optimaliseringsmethoden nog nauwelijks geïmplementeerd in multibody-analyse-software. Een belangrijke reden hiervoor is dat een evaluatie van doelfunctie en beperkingen meestal om een nieuwe oplossing van de bewegingsvergelijkingen vraagt. De koppeling tussen numerieke analyse en optimalisering wordt met name bemoeilijkt door het niet-lineaire en tijdsafhankelijke gedrag van de multibody-vergelijkingen. Voor grotere industriële toepassingen gaat dit vaak gepaard met hoge rekenkosten, hetgeen het aantal analyses dat door het optimaliseringsalgoritme uitgevoerd kan worden aan banden legt. Daarnaast wil een ontwerper ook niet graag geheel buiten het optimaliseringsproces gezet worden, hetgeen wél gebeurt als de multibody-analyse direct gekoppeld wordt aan het op-

timaliseringsalgoritme. Voor een succesvolle optimalisering is interactie met de ontwerper van belang, zodat ervaring en rekenkracht op een effectieve manier gecombineerd kunnen worden.

In dit proefschrift wordt voorgesteld om de koppeling tussen multibody-analyse en optimaliseringsalgoritme te realiseren door middel van benaderingsconcepten. Het benaderingsconcept dient dan als interface tussen numerieke analyse en optimalisering. Het principe is om benaderingsmodellen voor doelfunctie en beperkingen te genereren op basis van rekenresultaten in één of meerdere ontwerppunten. Het benaderend optimaliseringsprobleem kan vervolgens worden opgelost door het optimaliseringsalgoritme zonder dat een beroep gedaan hoeft te worden op de multibody-analyse. Er is aldus een scheiding aangebracht tussen analyse en optimalisering, waarbij de ontwerper mede zijn invloed kan laten gelden via de benaderingsmodellen. Aangetoond wordt dat deze manier van werken, die oorspronkelijk ontwikkeld is voor statische constructies, ook voor multibody-systemen goede mogelijkheden biedt om de numerieke optimalisering zowel toegankelijk als rekentechnisch beheersbaar te maken.

Het eerste deel van dit proefschrift beschrijft de optimaliseringsmethode. Uitgangspunt is de formulering van het optimaliseringsprobleem, met als karakteristiek element voor multibody-systemen de tijdsafhankelijkheid. Vervolgens wordt aangegeven hoe benaderingsconcepten hier kunnen worden ingepast. Een optimaliseringsgereedschap is ontwikkeld dat een benaderend optimaliseringsprobleem opstelt en oplost in een deelgebiedje van de ontwerpruimte, en vervolgens het zoekgebied verschuift in de richting van het berekende optimum. Dit leidt in het algemeen tot een herhaald benaderend optimaliseringsproces. Hierbij wordt onderscheid gemaakt tussen éénpuntsconcepten die naast de multibody-analyse ook een gevoeligheidsanalyse nodig hebben, en meerpuntsconcepten die geen afgeleiden naar de ontwerpvariabelen gebruiken. Voor de éénpuntsconcepten spelen nauwkeurigheid en efficiëntie waarmee de afgeleiden worden berekend een belangrijke rol. Daarom wordt met betrekking tot deze twee aspecten een overzicht gegeven van verschillende manieren waarop de gevoeligheidsanalyse voor multibody-systemen geïmplementeerd kan worden.

In het tweede deel wordt de praktische toepasbaarheid van de herhaald benaderende optimaliseringsmethode geïllustreerd aan de hand van drie ontwerp-problemen. Het meest eenvoudige probleem betreft een vierstangenmechanisme met flexibele staven. Uit dit voorbeeld blijkt dat dynamica de kwaliteit van de benaderingen kan beïnvloeden, en dat een geschikte keuze van de benaderingsmodelfuncties de convergentie van de optimalisering kan verbeteren. Een ander voorbeeld is de optimalisering van een gecombineerd airbag-gordel veiligheidssysteem. Kenmerkend is dat de veiligheidscriteria zich niet glad gedragen als func-

tie van de ontwerpvariabelen. Een meerpuntsconcept bewijst goed met deze extra ruis overweg te kunnen. Als derde applicatie wordt gezocht naar de optimale dempingskarakteristiek van een slagafhankelijke demper voor de voor-as van een truck. Verscheidene optimaliseringen zijn uitgevoerd gebruik makend van lineaire éénpuntsbenaderingen. Hier komt duidelijk naar voren dat ontwerp-optimalisering niet opgevat kan worden als een black-boxproces. De keuzes en sturing van de ontwerper zijn sterk bepalend voor het uiteindelijke resultaat.

Benaderingsconcepten zijn een bruikbaar hulpmiddel voor de numerieke optimalisering van multibody-systemen. Ze vormen een goede interface tussen de multibody-analyse en het optimaliseringsalgoritme, die de ontwerper een ingang geeft op het optimaliseringsproces. Dit biedt perspectieven om tot een interactief ontwerpgereedschap voor multibody-systemen te komen, waarin analyse en optimalisering geïntegreerd aanwezig zijn.

Curriculum Vitae

25 september 1969	Geboren te Hoorn
1981 - 1987	Gymnasium aan de scholengemeenschap Sint-Michiel te Geleen
1987 - 1992	Werktuigbouwkunde aan de Technische Universiteit Eindhoven
1992 - 1997	Assistent in Opleiding bij de vakgroep Fundamentele Werktuigkunde van de Technische Universiteit Eindhoven.

Dankwoord

Graag wil ik iedereen bedanken die op welke wijze dan ook een bijdrage heeft geleverd aan dit proefschrift. Ik ben dankbaar voor de ondersteuning die ik heb gekregen van mijn begeleiders, mijn (ex-)kamergenoten en collega's op de TU, en niet te vergeten de systeembeheerders. Met name wil ik noemen de studenten die in het kader van mijn promotieproject hun stage of afstuderen hebben uitgevoerd: Han Adriaens, Robert Borst, Twan van Etten, Johan de Jager, Richard Kolkhuis Tanke, Robert Rikken, Edwin Thijssen en Ruud Vermeulen. Zonder hen, en zonder de inbreng van Marc van Slagmaat (TNO) en Jos van Heck (DAF Trucks), had het proefschrift nooit zijn huidige vorm bereikt. Speciale dank gaat ook uit naar Saskia en mijn ouders, en naar mijn familie en vrienden.

Pascal Etman
Eindhoven, 18 juni 1997

Stellingen

behorende bij het proefschrift

Optimization of Multibody Systems using Approximation Concepts

1. Haug *et al.* (1984) alsmede Bestle en Eberhard (1992) stellen ten onrechte dat de gevoeligheidsanalyse dé ontbrekende schakel is tussen numerieke multibody-analyse en optimalisering.
 - Bestle, D. and Eberhard, P. (1992). Analyzing and optimizing multibody systems. *Mechanics of Structures and Machines*, **20**, 67-92.
 - Haug, E. J., Wehage, R. A., and Mani, N. K. (1984). Design sensitivity analysis of large-scale constrained dynamic mechanical systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **106**, 156-162.
 - Dit proefschrift, hoofdstuk 8.
2. Grootschalige computer-crash-modellen kunnen goed geoptimaliseerd worden met behulp van herhaalde meerpuntsbenaderingen van doelfunctie en beperkingen.
 - Dit proefschrift, hoofdstuk 6.
3. Als de gevoeligheidsanalyse lokaal een goede voorspellende waarde heeft, convergeert herhaald benaderend optimaliseren op basis van een éénpuntsconcept met afgeleiden beter dan op basis van een meerpuntsconcept zonder afgeleiden.
 - Dit proefschrift, hoofdstuk 3.
4. Het feit dat hulpvariabelen en hulpresponsvariabelen probleemspecifiek zijn, is juist voordelig, in plaats van nadelig zoals gesteld wordt door Barthelemy en Haftka (1993).
 - Barthelemy, J.-F. M. and Haftka, R. T. (1993). Approximation concepts for optimum structural design - a review. *Structural Optimization*, **5**, 129-144.
 - Dit proefschrift, hoofdstukken 3 en 5.

5. Het effect van de formulering van het optimaliseringsprobleem op het optimaliseringsproces en het uiteindelijk verkregen optimum wordt vaak onvoldoende onderkend.
6. Door te optimaliseren leer je een ontwerp pas echt goed kennen.
7. Onderzoek met betrekking tot evaluatie van bestaande methoden verdient meer aandacht.
 - Haftka, R. T. and Sobieski, J. (1992). Editorial: The case for helping consumers of research. *Structural Optimization*, **4**, 63-64.
8. Het maken van een goede instructie- of tentamenopdracht kost meer moeite dan het oplossen ervan.
9. Motivatie is de op één na belangrijkste factor die leren beïnvloedt.
 - The most important single factor influencing learning is what the learner already knows (Ausubel).
10. Wie altijd overal een mening over wil hebben, zal nooit tijd vinden zich echt een mening te vormen.

Pascal Etman

Eindhoven, 18 juni 1997