

Automated Approach for Optimizing Dynamic Systems

Dieter Bestle* and Peter Eberhard*

Abstract

The optimal design of nonlinear dynamic systems can be formulated as a multicriteria optimization problem. On the basis of a multibody system model integral type objective functions are defined evaluating the dynamic behavior of the system under consideration. Multicriteria optimization methods reduce the problem to nonlinear programming problems which can be solved with standard algorithms like the SQP method. The gradients required for such an efficient optimization procedure are computed by solving additional differential equations resulting from an adjoint variable approach. The whole design process can be highly automated by using computer algebra packages.

Introduction

Due to the complexity of technical systems and the wide variety of conflicting specifications for their dynamical behavior, dynamic systems have been designed by engineers with help of experience and intuition for a long time. The design process has been based on experimental studies of prototypes resulting in rather long development cycles due to their time-consuming and costly construction.

Only recently, production companies have started to switch to a computer-aided design process to shorten development cycles and improve their products. In most cases, however, computers are used for parameter studies only, whereas the design itself is still found by intuitive changes of the design variables. On the other hand, optimization algorithms for solving standard nonlinear programming problems are highly developed.

It is the aim of this paper, therefore, to describe an integrated modeling and design approach consisting of four phases [1]: (i) formulation of a mathematical model, (ii) choice of design variables, (iii) definition of criteria, and (iv) optimization. A multibody system approach will be used for generating models for complex dynamic systems. Parameters of the model will serve as design variables and two types of criteria will be defined. Finally, a multicriteria approach will be applied to account for the presence of conflicting performance criteria in applications to

*University of Stuttgart, Institute B of Mechanics, D-70550 Stuttgart, Germany

real systems. In an interactive design process, the design engineer can provide information on the importance of each criterion, and he can choose between several multicriteria methods for reducing the problem to one or a recursive sequence of nonlinear programming problems which are solved by SQP methods.

Formulation of the design problem

Computer-aided design of dynamic systems has to be based on mathematical models. If we can neglect small deformations of the individual parts, the multibody system approach has shown to be a good representation of the system. A multibody system model consists of rigid bodies connected by ideal links and coupled by ideal force elements like springs, dampers or actively controlled elements, Fig. 1. Multibody system models have been used with success in vehicle dynamics, robotics, satellite dynamics and biomechanics.

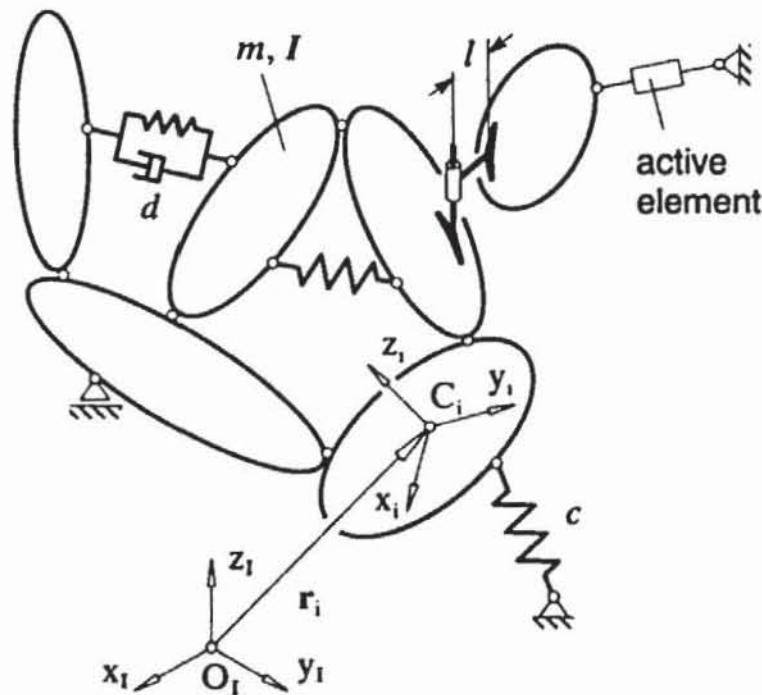


Figure 1: Multibody system model

Modeling technical systems as multibody systems involves an implicit parameterization. The dynamic behavior of the model is completely determined by parameters like the mass and moments of inertia of each body, geometrical dimensions, and damping and stiffness coefficients of coupling force elements. The parameters which can be changed within given ranges for optimizing the dynamical behavior are considered as design variables and summarized in a vector

$$\mathbf{p} \in \mathbb{R}^h, \quad p_k^l \leq p_k \leq p_k^u, \quad k = 1(1)h, \quad (1)$$

where p_k^l and p_k^u are lower and upper bounds, respectively, due to technical restrictions or physical meaning.

The dynamic behavior of a multibody system is described by differential equations of motion:

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{v}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) \\ M(t, \mathbf{y}, \mathbf{p}) \dot{\mathbf{z}} + \mathbf{k}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) &= \mathbf{q}(t, \mathbf{y}, \mathbf{z}, \mathbf{p}) \end{aligned} \quad (2)$$

where $\mathbf{y}(t) \in \mathbb{R}^f$ and $\mathbf{z}(t) \in \mathbb{R}^g$ are vectors of generalized coordinates and velocities, respectively, and f and g are the numbers of degrees of freedom for position and velocity, respectively. The equations of motion result from Newton's and Euler's laws and d'Alembert's or Jourdain's principle for eliminating reaction forces and moments, e.g. [9]. The mass matrix $M(t) \in \mathbb{R}^{g \times g}$ summarizes mass properties of the individual bodies, vector $\mathbf{k}(t) \in \mathbb{R}^g$ centrifugal and Coriolis forces, and $\mathbf{q}(t) \in \mathbb{R}^g$ applied forces resulting from gravity and coupling elements. For holonomic multibody systems we have $g = f$ and we can use $\mathbf{z} = \dot{\mathbf{y}}$.

The equations of motion for models of technical systems are already too complex for generating them by hand. Therefore, computer codes have been developed for a computer-aided modeling and generation of equations of motion in symbolical or numerical form [10]. Although such codes exist now for several decades, they are still subject of intensive research [11].

For a complete description of the motion, initial conditions for \mathbf{y} and \mathbf{z} have to be provided. This can be done by implicit conditions

$$\mathbf{y}^0 : \phi^0(t^0, \mathbf{y}^0, \mathbf{p}) = 0, \quad \mathbf{z}^0 : \dot{\phi}^0(t^0, \mathbf{y}^0, \mathbf{z}^0, \mathbf{p}) = 0 \quad (3)$$

for some fixed starting time t^0 .

In [2] the design problem has been stated for a single criterion. But generally, dynamic systems have to be optimal with respect to several specifications. Often such problems are simplified to nonlinear programming problems by choosing one criterion as objective function and the others as constraints. It is more natural, however, not to distinguish between objective functions and constraints in such an early design phase, and consider the decision on the importance and type of each criterion as part of a multicriteria optimization process.

Mainly, two types of criteria are used: we will call a criterion to be explicit if it is an algebraic function of the design variables:

$$\psi_i^E = \psi_i^E(\mathbf{p}), \quad i = 1(1)n_E. \quad (4)$$

A second type of performance criterion evaluating the dynamic behavior of multibody systems can be formulated as an integral type performance function

$$\psi_i^I = G_i^I(t^1, \mathbf{y}^1, \mathbf{z}^1, \mathbf{p}) + \int_{t^0}^{t^1} F_i(t, \mathbf{y}, \mathbf{z}, \dot{\mathbf{z}}, \mathbf{p}) dt, \quad i = 1(1)n_I, \quad (5)$$

which is also known from optimal control problems. The first term accounts for cases where special values for the final state $\mathbf{y}^1, \mathbf{z}^1$ or a minimum time t^1 must be

achieved, the second term evaluates the dynamic behavior within an interesting time interval $[t^0, t^1]$. The final time t^1 may be fixed or given implicitly by the final state:

$$t^1 : H^1(t^1, \mathbf{y}^1, \mathbf{z}^1, \mathbf{p}) = 0. \quad (6)$$

Although the functions G_i^1 and F_i depend on state variables, the functions ψ_i^1 are determined entirely by the values of the design variables \mathbf{p} due to Eqs. (2) and (3).

Multicriteria optimization

The problem of optimizing dynamic systems with respect to several conflicting criteria does not have a single optimal solution. The theory of multicriteria optimization has shown that the optimum depends on additional decisions of the designer.

At the beginning of the optimization phase, the designer has to classify all the criteria (4) and (5) as objective functions or constraints. Objective functions are criteria which should be minimized with respect to the design variables. Constraints are criteria which should have a special value or be less than an upper bound. Summarizing all objective functions in a vector function $\mathbf{f}(\mathbf{p}) \in \mathbb{R}^n$, all equality constraints in $\mathbf{g}(\mathbf{p}) = 0$, and all inequality constraints in $\mathbf{h}(\mathbf{p}) \leq 0$, we end up with the optimization problem

$$\underset{\mathbf{p} \in \mathcal{P}}{\text{minimize}} \mathbf{f}(\mathbf{p}) \quad \text{where} \quad \mathcal{P} := \left\{ \mathbf{p} \in \mathbb{R}^h \mid \mathbf{g}(\mathbf{p}) = 0, \mathbf{h}(\mathbf{p}) \leq 0 \right\}. \quad (7)$$

If only a single criterion is left, $n = 1$, problem (7) is called a nonlinear programming problem [4]. More realistic, however, is that more than one criterion has to be minimized simultaneously, $n > 1$. Then it is called a multicriteria or vector optimization problem, e.g. [8].

In the latter case, we cannot expect a feasible design point $\mathbf{p} \in \mathcal{P}$ where all objectives become minimal. Therefore, a design point $\mathbf{p}^P \in \mathcal{P}$ is defined to be Pareto-optimal if there is no other feasible point \mathbf{p} with $f_i(\mathbf{p}) \leq f_i(\mathbf{p}^P) \forall i$ and $f_j(\mathbf{p}) < f_j(\mathbf{p}^P)$ for at least one j [13]. In general, Pareto-optimal solutions are not unique, and the designer has to choose a special Pareto-optimal point as desired solution due to additional information on the design problem. For finding such points the multicriteria optimization problem has to be reduced to a scalar one for which efficient routines exist, Fig. 2. This reduction is based on two principles: scalarization and hierarchization.

In case of scalarization, Fig. 3a, the objective functions are combined to a new utility function $u(\mathbf{p}) \in \mathbb{R}$ which will be optimized instead of the vector criterion. A well known approach using scalarization is the weighting objectives method:

$$u(\mathbf{p}) := \sum_{i=1}^n w_i \frac{f_i(\mathbf{p})}{\hat{f}_i}, \quad \sum_{i=1}^n w_i = 1 \quad (8)$$

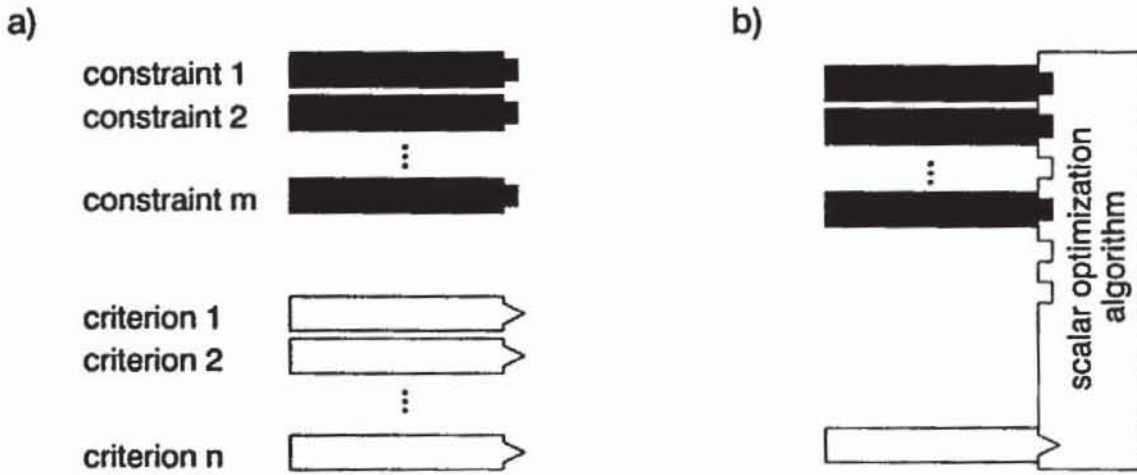


Figure 2: Difference between vector optimization problems (a) and scalar optimization problems (b)

where $w_i \in [0, 1]$ are weighting coefficients and \hat{f}_i are scaling factors. Other possibilities are absolute and relative distance functions with respect to a predefined design point [5].

For hierarchical methods the designer has to assign a level of importance l_i to each objective function $f_i(\mathbf{p})$ where one is the level of most important criteria. Especially, level zero is assigned to the equality and inequality constraints in Eq. (7), i.e.,

$$\mathcal{P}_0 := \mathcal{P} \quad \longleftrightarrow \quad l_0 = 0. \tag{9}$$

If there is only a single criterion on each level, we define in the first step a scalar optimization problem by neglecting the objectives on lower levels, and taking into consideration only the objective function on level one and the constraints, Fig. 3b:

$$f_i^* = \min_{\mathbf{p} \in \mathcal{P}_0} f_i(\mathbf{p}) \quad \text{where } i : l_i = 1. \tag{10}$$

For the next step we can use the information on the optimal value f_i^* of the most important objective to define a constraint on $f_i(\mathbf{p})$:

$$\mathcal{P}_i := \left\{ \mathbf{p} \in \mathbb{R}^h \mid f_i(\mathbf{p}) \leq (1 + \varepsilon_i) f_i^* \right\} \tag{11}$$

where $\varepsilon_i > 0$ is a user defined tolerance for function increase. Then we formulate a new scalar optimization problem for the objective function on level two similar to Eq. (10). The whole procedure is a recursive sequence of n scalar optimization problems:

$$f_i^* = \min_{\mathbf{p} \in \bigcap_{k: l_k < j} \mathcal{P}_k} f_i(\mathbf{p}) \quad \text{where } i : l_i = j, \quad j = 1, 2, \dots \tag{12}$$

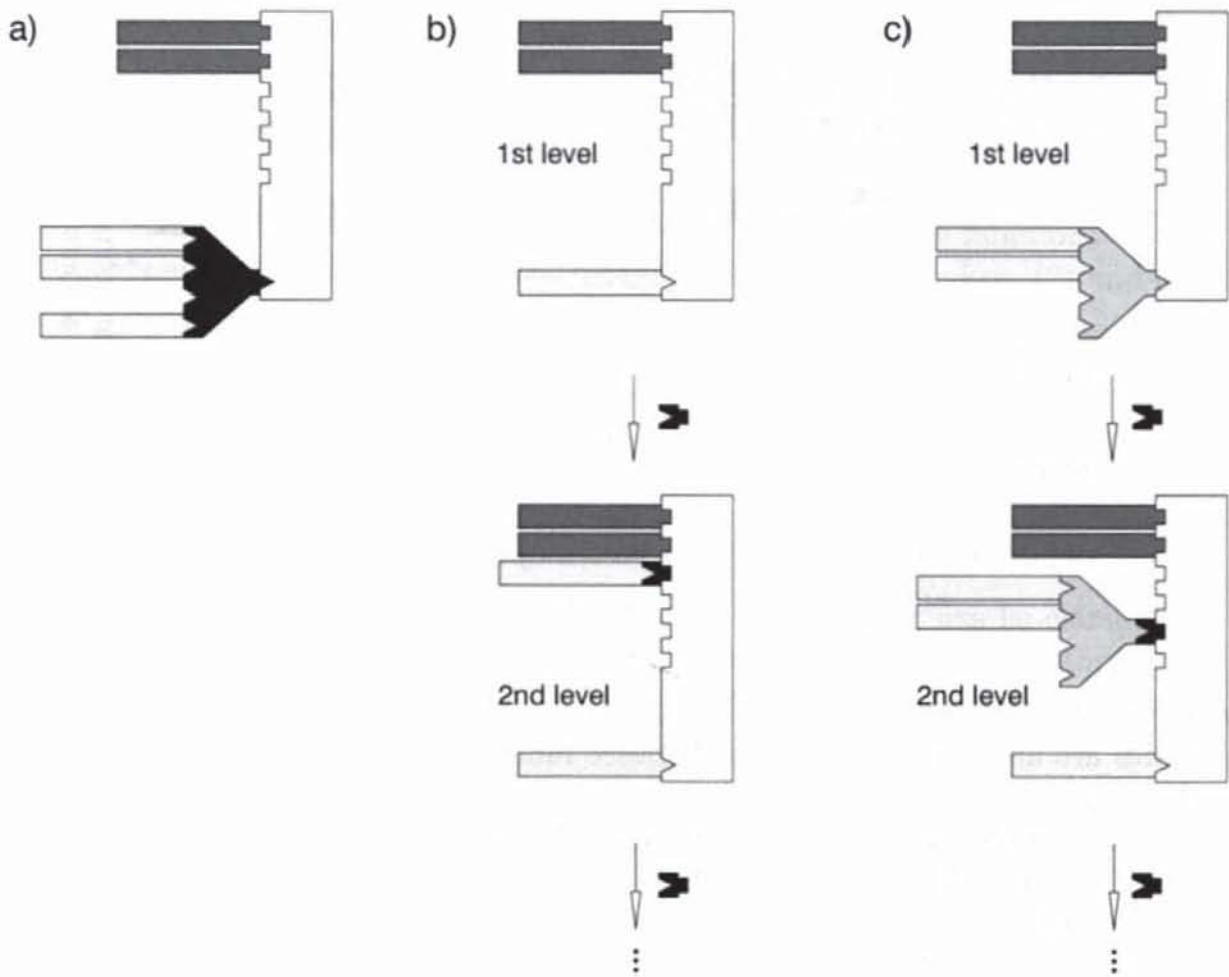


Figure 3: Multicriteria optimization principles: (a) scalarization, (b) hierarchization, and (c) combination

The scalarization and hierarchization principles can be combined if several objectives are on the same level, Fig. 3c. For example, goal programming becomes a flexible tool and gives good insight into the problem if both principles are used together. The user has to define goals \hat{f}_i to be reached and assign levels l_i to each objective $f_i(\mathbf{p})$. Then, instead of the objectives the deviations from the goals are minimized. Objectives on the same level j can be combined to utility functions similar to Eq. (8):

$$u_j(\mathbf{p}) := \sum_{i: l_i=j} w_i \max\{0, f_i(\mathbf{p}) - \hat{f}_i\}, \quad \sum_{i: l_i=j} w_i = 1. \quad (13)$$

The utility functions can then be handled according to Eqs. (10) to (12).

Solution of the scalar optimization problem

The scalar optimization problems resulting from a multicriteria approach have themselves to be solved in an iterative procedure. Due to the high computational

effort for evaluating integral type performance functions by numerical integration, optimization algorithms like the SQP methods with nice convergence properties should be used. The drawback of such methods, however, is the use of gradients.

A simple way of computing gradients is the use of finite differences. Applied to integral type criteria these approximations cause several problems. Numerical experience has shown that due to the limited accuracy of the function values finite differences are not very reliable near the optimum. On the other hand, we need one additional function evaluation for each perturbed design variable which is a time consuming numerical simulation of the dynamic behavior of the multibody system.

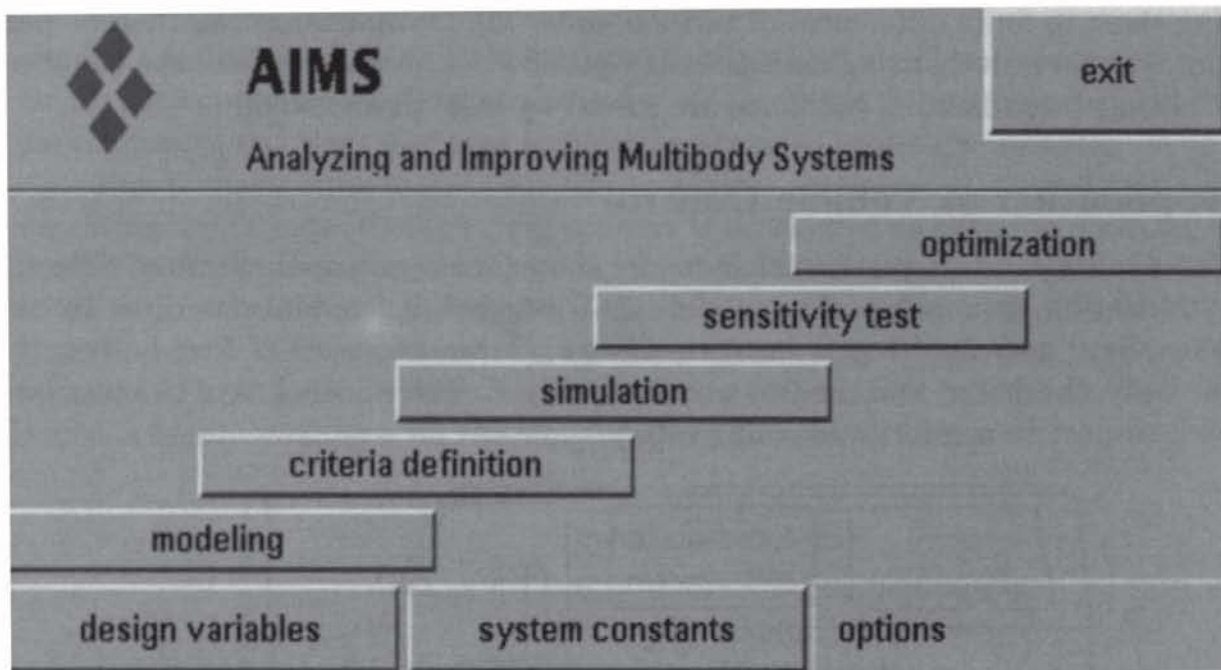


Figure 4: Graphical user interface AIMS

Therefore, a semi-analytical approach is used for computing gradients of this type of criteria which is called adjoint variable method [2]. This results in additional differential equations for the gradient where the computational effort is almost independent of the number of design variables. Numerical studies have shown high reliability and about the same accuracy for the gradients as for the function values.

Automated optimization approach

Due to the complexity of models for technical systems the design process has to be supported by approved computer programs. These programs can be integrated in a graphical user interface which also helps to organize the whole design process and especially the interactive and iterative optimization phase, Fig. 4. The user interface AIMS (Analyzing and Improving Multibody Systems) integrates several

numerical and computer algebra programs for modeling, simulation and optimization.

Modeling is supported by the computer program NEWEUL [7] which generates symbolical equations of motion for multibody systems. Criteria of type (4) or (5) have to be defined by the user in a MAPLE-compatible form. The computer algebra package MAPLE [3] will then generate problem-specific INCLUDE-files which can be linked together with problem-invariant FORTRAN-code for simulation and optimization. Simulation is performed with a multistep integration algorithm [12]. As already mentioned, the gradients are computed from additional differential equations which can also be generated by MAPLE. These gradients should always be checked on consistency with the problem definition by comparing them to finite differences of variable order [2]. Optimization can then be performed interactively using multicriteria optimization methods where the resulting nonlinear programming problems are solved by SQP methods [6].

Application to Vehicle Control

The application to a plane vehicle model shows some principal effects of different optimization approaches. The model has 6 degrees of freedom described by the generalized coordinates $\mathbf{y} = [y, z, \alpha, \phi, w, z_D]^T$, and consists of four bodies: the car body, the driver, and the two wheel sets, Fig. 5. The vehicle has to be optimized with respect to comfort and riding safety.

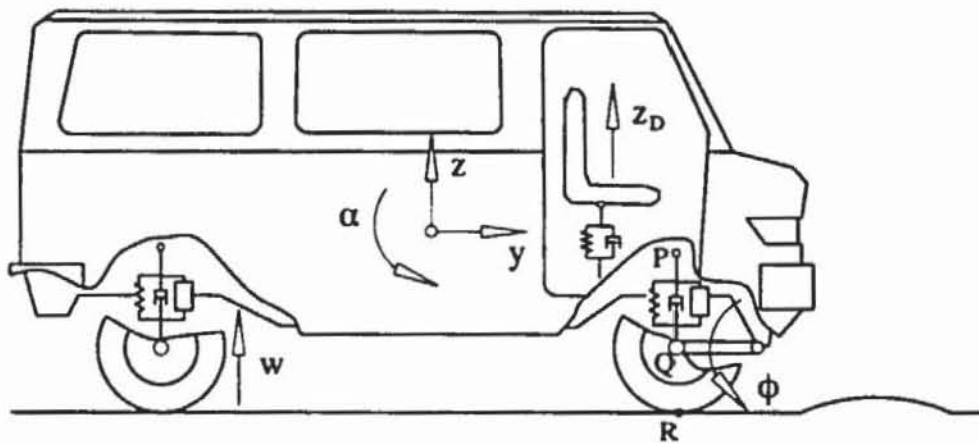


Figure 5: Plane vehicle model

A frequently used measure for comfort is the vertical acceleration \ddot{z}_D of the driver. Since driving over a bump is considered as a test, accelerations are penalized by time to avoid long term vibrations:

$$f_1 := \int_{t^0}^{t^1} (t \ddot{z}_D)^2 dt. \quad (14)$$

Optimal comfort is then expressed by a minimal value of f_1 . Riding safety is related to the dynamic variation of the load between the wheels and the road. If the tire is

considered as a linear spring, the load is proportional to the relative displacement between the wheel and the road surface:

$$f_2 := \int_{t^0}^{t^1} (z_Q - z_R)^2 dt. \quad (15)$$

A constraint on the design of suspension systems is the limited space for relative displacement between wheel and car-body. A criterion like

$$f_3 := \int_{t^0}^{t^1} \left(\frac{z_P - z_Q}{s_0} \right)^6 dt \quad (16)$$

may be used where s_0 is a predefined amplitude which should not be exceeded too much. For improving the dynamic behavior of the vehicle the stiffness and damping parameters of the front and rear suspension are used as design variables.

Fig. 6 shows some results for the weighting objectives method for different weighting coefficients. If only riding comfort is considered as criterion it can be improved drastically compared to the initial design. But this improvement is achieved at the expense of riding safety and it requires a large suspension displacement. For a more realistic design optimization all three criteria have to be taken into consideration. As Fig. 6 shows, riding comfort still can be improved but the improvement depends highly nonlinear on the weighting coefficients.

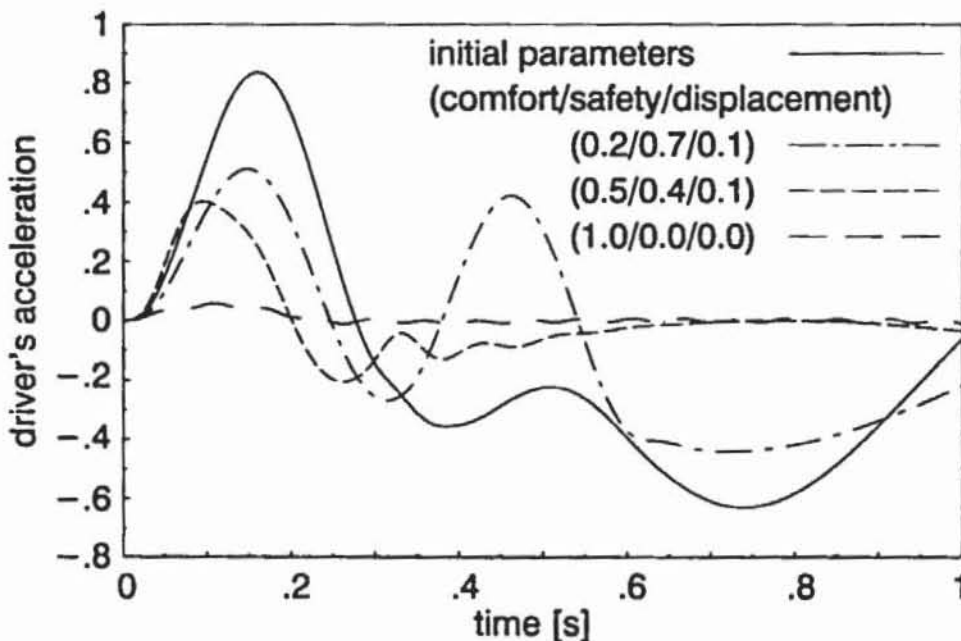


Figure 6: Weighting objectives method

Goal programming can give better insight into the problem. As a measure of importance, level one is assigned to riding comfort, level two to riding safety, and level three to relative displacement. Fig. 7 shows the results of three different

runs where the values of the criteria are normalized with respect to the initial design. In the first run, the goals are set very cautiously and can be achieved. In the second run, the goal for riding safety is decreased and cannot be achieved anymore. Therefore, in the third run the goal for riding safety is increased and the relative displacement can be decreased at the expense of comfort. In a practical design process, the designer has to explore the design space with further runs and find a suitable balance for the conflicting criteria.

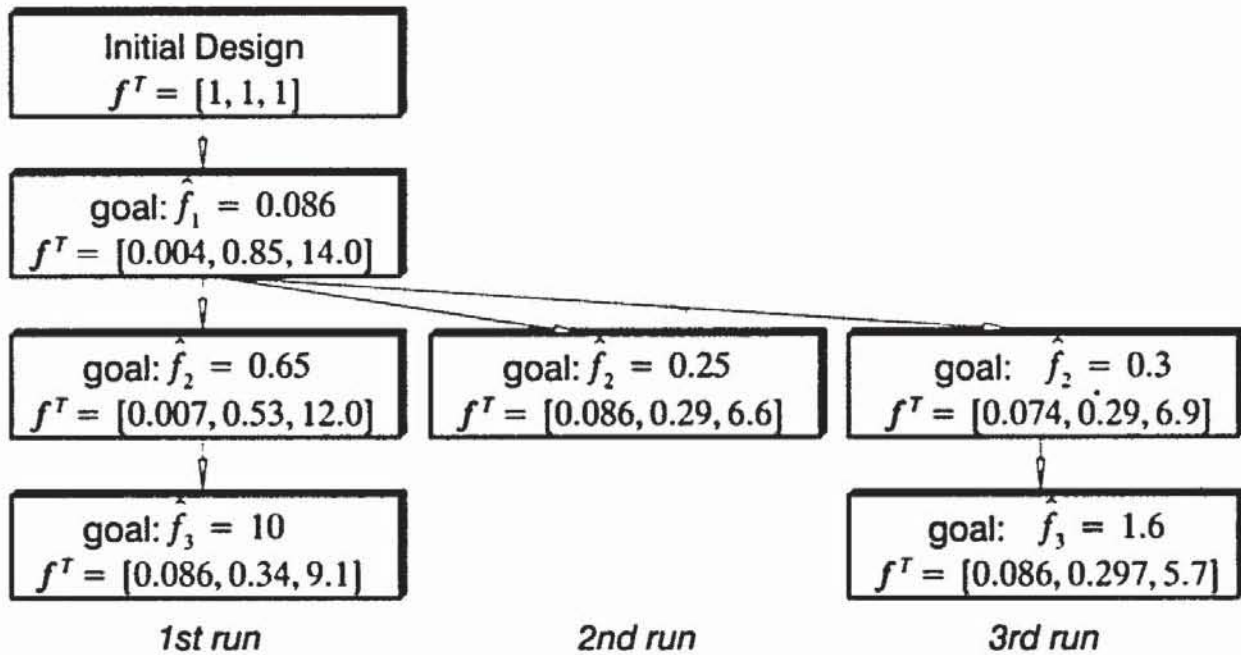


Figure 7: Goal programming method

References

- [1] BESTLE, D., Analyse und Optimierung von Mehrkörpersystemen, to appear.
- [2] BESTLE, D., AND EBERHARD P., Analyzing and Optimizing Multibody Systems, Mech. Struct. and Mach. **20** (1992) 67–92.
- [3] CHAR, B.W., ET. AL., MAPLE-Reference Manual, Waterloo Maple Publ., Waterloo 1990.
- [4] FLETCHER, R., Practical Methods of Optimization, Wiley, Chichester 1987.
- [5] HWANG, C.-L., AND MASUD, A.S., Multiple Objective Decision Making: Methods and Applications, Springer, Berlin 1979.
- [6] IMSL MATH LIBRARY, User's Manual, IMSL Inc., Houston 1989.
- [7] KREUZER, E., AND LEISTER, G., Programmsystem NEWEUL'90, Anleitung AN-24, Institute B of Mechanics, University of Stuttgart, Stuttgart 1991.
- [8] OSYCZKA, A., Multicriterion Optimization in Engineering, Ellis Horwood, Chichester 1984.

- [9] SCHIEHLEN, W., Technische Dynamik, Teubner, Stuttgart 1986.
- [10] SCHIEHLEN, W. (ed.), Multibody Systems Handbook, Springer, Berlin 1990.
- [11] SCHIEHLEN, W. (ed.), Advanced Multibody System Dynamics, Kluwer, Dordrecht 1993.
- [12] SHAMPINE, L.F., AND GORDON, M.K., Computer Solution of Ordinary Differential Equations: The Initial Value Problem, Freeman, San Francisco 1975.
- [13] STADLER, W., Multicriteria Optimization in Mechanics: A Survey, Appl. Mech. Rev. **37** (1984) 277–286.