1 INTRODUCTION

Carbon dioxide (CO_2) emissions in Europe are required to be reduced from 130 to 95 g/km in the period from 2015 to 2020 in accordance with the objectives of the European Commission. [1] Statistics show that the average emission level of a new car sold in 2014 was 123.4 g/km. [1] To meet the regulations by 2020, 24% of CO_2 has to be reduced in five years; this is a challenging target for car manufacturers.

Rolling resistance contributes to 25% of the total fuel consumption in automobiles, which indicates that there exists considerable potential for the reduction of CO_2 emissions by reducing the rolling resistance of a tire. [2] A consideration of the properties of a chassis reveals that a reduction in the rolling resistance of a tire is important for reducing CO_2 emissions.

Several tire properties such as traction or grip, longevity, fuel efficiency, handling, comfort, noise, and robustness have to be considered for the purpose of attempting to reduce the rolling resistance of a tire. A tire is a complex system in which the characteristics of each of the abovementioned properties can be varied depending on the designer's intention. If one property is significantly improved or changed, then a trade-off takes place with other properties. This means that the reduction in the rolling resistance of a tire may result in some disadvantageous tire characteristics, such as lower cornering stiffness and longer relaxation length. Irrespective of the changes in the properties of the tires, the overall characteristics of the vehicle should be maintained based on predefined requirements.

Different cornering stiffnesses and relaxation lengths lead to different lateral dynamics in a vehicle. From the handling point of view, a vehicle with a lower cornering stiffness at the front and rear tends to exhibit sluggish lateral dynamics. Therefore, this effect should be compensated for through chassis optimization, and thus, the aim of this study is to obtain the optimum configuration of the chassis. For instance, a lower cornering stiffness of the tire can be compensated for with more roll steer or greater lateral compliance steer, which enables the buildup of a larger sideslip angle of nearly the same extent of side forces as in a reference vehicle. To some extent, some characteristics of the tire can be compensated for through chassis manipulation, and the corresponding extent of alteration in the cornering stiffness and relaxation length under realistic boundary conditions is discussed in detail. This chassis optimization process can be treated as a special case involving rolling-resistance optimized tires (RR tires).

For the chassis optimization process, the characteristic-curve-based double-track model has been used. Using these characteristic curves, the number of parameters required to define a chassis configuration can be minimized. Certain characteristic curves can be altered without modifying the other characteristic curves. The influence of each characteristic curve on the vehicle dynamics can be better understood. In addition, all possible chassis design variables, not only kinematics but compliance also, can be considered with fewer optimization parameters. Regardless of this advantage, there exists a problem of whether the obtained chassis configurations can be realized by varying the actual chassis setup, e.g., the position of the hardpoints and stiffness of the bushings. In order to eliminate unrealistic chassis configurations, boundary conditions have been considered during the optimization process.

The main focus is on which characteristic values should be used to evaluate the lateral dynamics of a vehicle. In this study, an overshoot and a peak time of the sideslip angle for a step steer maneuver at 150 km/h with two yaw gains at 80 and 150 km/h has been used for the evaluation. The objective functions are applied for the evaluation of the vehicle, and each sub-objective function is multiplied by its corresponding weighting factors. It is also explained how the weighting factors have been objectified to realize certain characteristic values. The relevant lateral dynamics characteristic values must achieve the target on priority in comparison to the other characteristic values, and therefore, it is very important to use objectified weighting factors.

In Chapter 2, the modeling used for the simulation is introduced. In Chapter 3, the applied method and boundary conditions, the formulation of the objective functions, the modifications of the chassis considered for the optimization, and the driving maneuvers to be used are presented. In Chapter 4, the characteristic values to be considered for evaluating the lateral dynamics of the vehicle are discussed. In Chapter 5, the results of the design of experiments are presented to define a correlation between the chassis configurations and the characteristic values. Chapter 6 describes how the weighting factors are derived in order to realize the relevant characteristic values for the lateral dynamics while minimizing the worsening of the other characteristic values. The optimization results are presented in Chapter 7, and they provide an overview of whether the lower cornering stiffness and longer relaxation length can be thoroughly compensated for.

1.1 STATE OF THE ART

With respect to the handling characteristics of a vehicle, there are various handling objectives that are evaluated as characteristic values based on driving maneuvers. These handling objectives can be treated in a manner similar to multi-objective optimization problems. The methodologies used to solve such a multi-objective optimization problem differ based on the variations in the chassis system, formulation of the sub-objective function, distribution of the weighting factors, and whether an additional model such as neural network (NN) is used for estimating the result.

Focusing on single multi-criteria optimization, Gobbi et al. [3] dealt with ride and handling optimization using two different mathematical models. The first mathematical model is a physical model that is strictly related to the actual vehicle, and the second model is an approximation model that is just a set of mathematical functions that are capable of approximating the different relationships existing between the design variables and the characteristic values. "Global approximation" is the substitution of a physical model for a numerical model and allows a quick computation of the Pareto-optimal set. [4] A genetic algorithm is used to obtain a direct computation of the Pareto-optimal set and a trained artificial neural network (ANN) is used during the search procedure. However, this methodology requires an extensive number of function evaluations to obtain a sufficiently trained representative ANN. Secondly, the kinematic and compliance of the suspension were not considered but only simple parameters such as the stiffness and damping of the suspension, stiffness of the anti-roll bar, and toe and camber angles are considered for the optimization in the paper.

Schuller et al. [5] carried out an optimization of vehicle-handling behavior using a simulation; the characteristic-curve-based double-track model and a genetic algorithm were used for the optimization. In comparison to Gobbi et al., the kinematic and compliance of the suspension are considered as the optimizing parameters. Nevertheless, no restrictions are made with respect to the suspension design, i.e., the characteristic curves of the suspension were optimized without considering their feasibility in real conditions. A coupling between the suspension variables is neglected, and this leads to restricted reliability of the optimization results.

Benedetti et al. [6] presented a method of fuzzy optimality and k optimality, that dealt with a large number of design variables and objectives. This methodology requires differential modeling, NN modeling, design of experiments, statistical correlation analysis, and evolutionary multi-objective optimization. The number of objective functions is reduced by carrying out a correlation analysis of the objective functions, while excluding objective functions that correlate the other objective functions that are already in consideration. Because NN modeling is used, this methodology also requires extensive training to improve the accuracy of the NN model. The kinematic and compliance characteristic curves were not considered as design parameters for the optimization, which simplifies the optimization problem. Nevertheless, this study presents a method for reducing the number of required objective functions by means of correlation analysis.

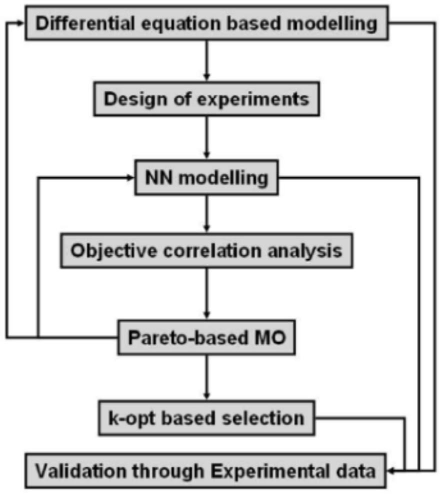

Figure 1.1 Block diagram of k optimality. [6]

Thoresson et al. [7], [8] compared the gradient-based optimization results regarding handling and ride comfort using a multibody simulation model and a simplified model in MATLAB. They concluded that using the simplified model in MATLAB facilitates lower computational cost and numerical noise.

Angrosch et al. [9] aimed to determine whether the numerical optimization and design of experiments are applicable to the design of suspension systems. A multibody simulation model is used in this study, and it shows a direct relation between the position of the hardpoints and the values of the vehicle handling characteristics. For the analysis, a McPherson suspension was used for the front and a multi-link suspension was used for the rear. In this study, only the variations in the hardpoints have been considered. When considering the stiffness of the bushings, the number of design parameters increases dramatically, which makes the optimization more difficult.

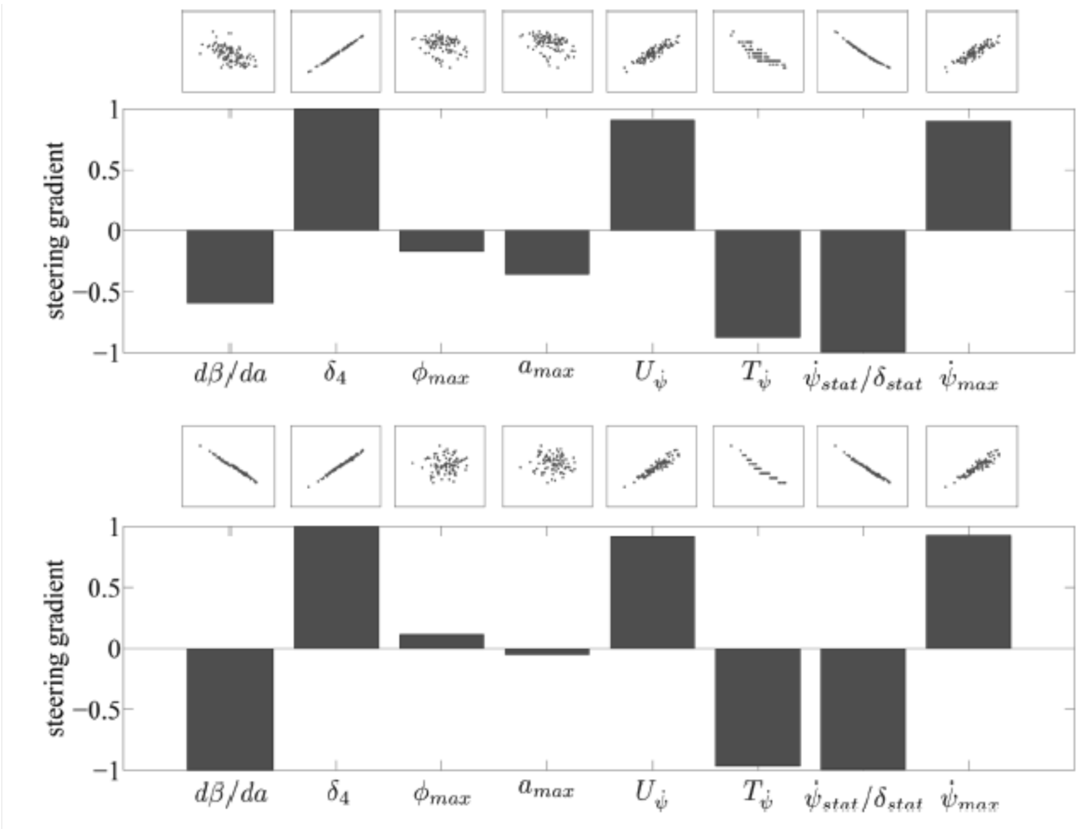

Figure 1.2 Variation of the hardpoints at front suspension. [9]

In all the aforementioned studies, either the chassis variation was not detailed enough to consider all possible chassis configurations or an additional model such as an NN model was considered, for which a large number of simulations are required for improved the accuracy. Moreover, the main problem of neural networking is that it is just a high probability estimation and optimization result, and using such a model does not guarantee the same result in the real physical model.

1.1.1 Optimization Algorithms

There are several different optimization algorithms [10]–[13] available and they can be categorized as gradient-based or non-gradient-based algorithms. An example of a non-gradient-based optimization algorithm is a genetic algorithm, which is commonly used to solve nonlinear problems. Only objective function evaluations are used to find the global optimum, and therefore, more time is required to solve the problem. In contrast, gradient-based optimization algorithms require the existence of continuous first derivatives of the objective function and possibly higher derivatives as well. Such algorithms converge to the optimum relatively quickly, but only the convergence to local minimum is guaranteed. Methods that are frequently used for gradient-based algorithms are as follows:

- Trust-region-reflective algorithm [14], [15]
- Sequential quadratic programming algorithm [16], [17]

Consider a constrained optimization problem such as

$$\begin{gathered}\text{minimize } f(X) \\ X \in \boldsymbol{R}^n\end{gathered} \tag{1.1}$$

with the boundary condition of $l \leq X \leq u$; where u and l are respectively the vectors of the upper and lower boundary conditions of a vehicle.

Trust Region Methods for Nonlinear Minimization

This is a simple but powerful algorithm. In the problem mentioned in Equation (1.1), where the function takes vector arguments (X) and returns scalars, the basic idea is to approximate f with a simpler function q, which reflects the behavior of function f in a neighborhood N around the point X. This neighborhood is the trust region. A trial step s is computed by minimizing the simpler function q over the region of trust (N).

$$\min \{q(s), s \in \mathrm{N}\} \tag{1.2}$$

The current point (X) is to be updated if $f(X + s) < f(X)$ is fulfilled; otherwise, the current point remains the same, the region of trust (N) becomes smaller, and the trial step is repeated. The main problem would be how to choose and compute the approximation (defined at current point x), how to choose and modify the trust region N, and how to accurately solve the trust region sub-problem depicted in Equation (1.2). In a standard trust region method, q is defined as the first two terms of the Taylor approximation of f at X. The sub-problem can then be formulated as follows: [18]

$$\min \left\{\frac{1}{2} s^T H s + s^T g \text{ such that } \parallel Ds \parallel \leq \Delta\right\}, \tag{1.3}$$

where g is the gradient of f at the current point X, H is the Hessian matrix, D is a diagonal scaling matrix, Δ is a positive scalar, and $\|.\|$ is the 2-norm. Solving Equation (1.2) using algorithms employing the computation of a full eigensystem requires time proportional to the factorization of H. In order to simplify the problem, the trust region sub-problem is treated as a two-dimensional subspace S. S is the linear space spanned by s_1 and s_2, where s_1 is in the direction of the gradient g, and s_2 is either an approximate Newton direction (solution to

$H \cdot s_2 = -g$) or a direction of negative curvature (solution to $s_2^T \cdot H \cdot s_2 < 0$). [19] Unconstrained minimization can be performed by formulating the two-dimensional trust region sub-problem (S), solving Equation (1.2) to determine the trial step s if $f(X+s) < f(X)$, then $X = X + s$, and, finally, adjusting Δ. These steps are repeated until convergence.

Sequential Quadratic Programming (SQP)

A quadratic programming (QP) sub-problem based on a quadratic approximation of the Lagrangian function (L) should be formulated.

$$L(\mathrm{x}, \lambda) = f(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x) \quad (1.4)$$

The QP sub-problem is defined as follows:

$$\begin{gathered} \min\left(\frac{1}{2} d^T H_k d + \nabla f(x_k)^T d\right) \\ \nabla g_i(x_k)^T d + g_i(x_k) = 0, i = 1, \ldots m_e \\ \nabla g_i(x_k)^T d + g_i(x_k) \leq 0, i = m_e, \ldots, m. \end{gathered} \quad (1.5)$$

This sub-problem can be solved using a QP algorithm and its solution forms a new iterate:

$$x_{k+1} = x_k + \alpha_k d_k \quad (1.6)$$

The step length parameter (α_k) is determined by an appropriate line search procedure such that a sufficient decrease in a merit function can be obtained, where d_k is the search direction, and λ_i is the Lagrange multiplier. The matrix H_k is a positive definite approximation of the Hessian matrix of the Lagrangian function. H_k can be updated using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method.[20] Studies [21]–[26] have investigated how the global convergence can be pursued.

Genetic Algorithm

A genetic algorithm [27]–[34] is commonly used to find a global minimum in highly nonlinear problems; only the objective function evaluations are considered to find the optimum point. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. A genetic algorithm works using the following sequence: [35], [36]

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
    a. The objective function value of each individual in the current population is evaluated by computing its multi-objective function value.
    b. The raw objective function evaluation is scaled to obtain a more usable range of values.
    c. Members, called parents, are selected based on their objective function value.
    d. Some of the individuals in the current population that have a lower objective function value are chosen as elite. These elite individuals are passed to the next population.
    e. Children are produced from the parents either by making random changes to a single parent (called mutation) or by combining the vector entries of a pair of parents (called crossover).
    f. The current population is replaced with the children to form the next generation.
3. The algorithm stops when one of the stopping criteria is met.

1.2 REQUIREMENTS FOR THE NEW METHODOLOGY

In summary, the following are the requirements for the chassis optimization.

1. To avoid conflicts between objective functions as much as possible, the number of optimization criteria should be kept to a minimum.

2. The optimization result should reflect the reality regarding the boundary conditions. In other words, feasible boundary conditions should be considered during the optimization.
3. It would be preferable to deal with multi-objective functions without using an NN.
4. All possible chassis design variables, not only kinematic but also compliance, should be considered during the optimization.
5. The optimum obtained using an optimizer should be a global optimum.

An optimization of the characteristic curves of the double-track model is presented in this study. The characteristic curves to be manipulated with a constant parameter are required to be predefined. Each chassis design parameter represents a certain modification of the chassis, and a set of these parameters produce a vector. With this vector regarded as an input and the corresponding value from the objective function regarded as an output, the genetic algorithm is used to perform the optimization, and its results are dependent on the following factors:

- chassis modifications (what is modified) and their limits
- driving maneuvers and their characteristic values
- weighting factors in objective functions
- selected optimization algorithm: genetic algorithm

All possible chassis modifications are considered, and the boundary conditions of each measure are maintained as broad as possible to take all possible configurations into account. The boundary conditions are decided based on the feasibility of the subsystem and the component levels. The optimization is conducted such that each set of vectors within the predefined boundary conditions can always be realized at the subsystem and component levels. The optimization in this study mainly focuses on the entire vehicle and system levels.

Selecting an appropriate optimization engine to obtain a global minimum of the objective function is a subject of considerable debate in the field of optimization. Angrosch et al. [9] presented a table that compares different algorithms in terms of duration and objective function; this table indicates that a genetic algorithm shows the best result with a longer duration in comparison with SPQ, the method of feasible directions, and the adapted response surface method. However, it should be kept in mind that none of the aforementioned algorithms guarantee that a global optimum will be obtained.

The duration of an optimization and its accuracy in using the genetic algorithm varies depending on the settings of convergence termination, population size, initial population, and mutation rate of the genetic algorithm. The time required for the optimization using a genetic algorithm has been shortened using parallel computing. Using eight multiple processors in parallel, the vehicle maneuver of eight individuals can be evaluated simultaneously.

1.3 V-MODEL AND TARGET CASCADING

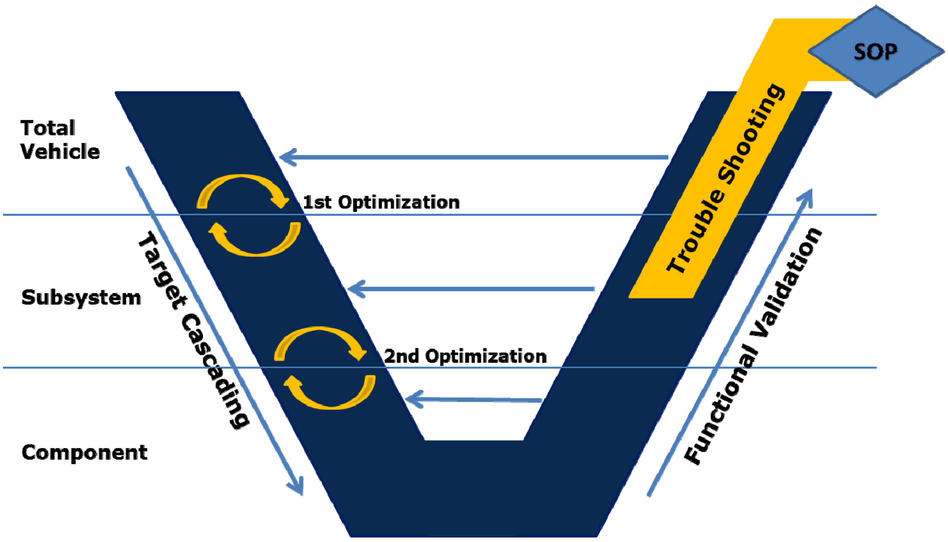

Figure 1.3 Application of V-model in the vehicle development process. [37]

Figure 1.3 shows a V-model and provides an overview of the vehicle development process. Starting with the requirements of the desired full vehicle characteristics, the downward movement in the left side of the 'V' represents decomposition and specification from the requirements of the full vehicle to the design and simulation of the system and finally to the design and evaluation of the components and parts. The systems that are created using the designed components are tested and validated against their specifications in hierarchical order, i.e., from components over subsystems to the full vehicle. The system design and validation takes place at the same level in the V-model. [37]

Target cascading in product development consists of a systematic effort to propagate the desired top-level system design to appropriate specifications for subsystems and components in a consistent and efficient manner. The target cascading process attempts to achieve this consistency and concurrency early in the development process. The important specifications or "targets" for the entire system (as well as for each subsystem and component), specifically those that will influence other parts of the system, are identified first. [38]