FE-Simulation Based Optimization of an Adaptive Restraint System Considering Multiple Front-Crash Load Cases using LS-OPT

Marcel van den Hove*
Bernd Mlekusch*
Heiner Müllerschön**

* AUDI AG - www.audi.de
Ingolstadt, Germany
marcel.vandenhove@audi.de
bernd.mlekusch@audi.de

** DYNA*more* GmbH - www.dynamore.de
Stuttgart, Germany
hm@dynamore.de

Summary

The purpose of this paper is to explore some interesting aspects of optimization for crashworthiness occupant safety applications and to propose optimization strategies for highly nonlinear problems. With the today's state of technology it is possible to identify specific load cases and different types of occupants in the car. System parameters of the restraint system, such as trigger time for seat-belt, airbag and steering column can be adapted to particular load cases. This is refered to an adaptive restraint system.

In the first part of the paper different optimization strategies are discussed and pros and cons are compared. In addition, a methodology to get a reliable surrogate model using neural networks is introduced. The surrogate model (*Meta-Model* or *Response Surface Model*) approximates the relationship between design parameters and a physical response and can be used to visualize and explore the design space.
In the second part the application of the *Successive Response Surface Scheme* (SRSM) for the optimization of an adaptive restraint system is conducted. For this, several front crash load cases are considered. This is performed using LS-OPT (*Stander et al.* [11]) as optimization software and PAM-Crash as solver for the finite element occupant safety simulations.
The procedure of generating an advanced meta-model to get an approximation of the global design space using neural networks is demonstrated for this example. Furthermore, the visualization of multi-dimensional meta-models in two- and three-dimensional design space is illustrated.

Keywords

Passive Safety, Adaptive Restraint System, Crashworthiness Optimization, LS-OPT, Successive Response Surface Method (SRSM), Neural Network

1 Introduction

Nowadays restraint systems in passenger cars are highly sophistcated set-ups composed of units such as adaptive airbags, seat belt systems and steering wheel components. These components are controlled by several parameters, which have to be adapted in order to improve and optimize the safety performance of the vehicle. There are system parameters, which are fixed values, for example diameter of airbag venthole, and there are control parameters, which define a specific time for the activation of units such as for example time of deployment of an airbag. In the later case the trigger times are realized by sensors and can be adapted individually to specific load cases. It can be distinguished between heavy and light passengers or between belted and unbelted persons.
Many of these parameters have a strong influence on the injury performance of the passengers. There are highly non-linear relations between parameters and responses, in some cases even discontinous effects occur, e.g. if the dummy-head strikes through the airbag and hits onto the steering wheel. In this case the acceleration values suddenly jump to exorbitant values.
For the engineer it is almost impossible to understand and manage these causal relationships without an appropriate optimization tool.
The question arises which optimization methodology should be used for such highly nonlinear problems. This question is discussed in Section 2 and several approaches are compared.

2 Optimization Strategies for Highly Nonlinear Applications

Highly nonlinear applications can be defined as computations whith strong variations in response values, due to large structural deformations, bifurcations (folding, buckling) resulting in multiple solution branches, difficult contact conditions or sensitive systems with successional events, which relate to each other.

Examples for highly nonlinear applications are:

- Crashworthiness Computations
- Occupant/Pedestrian Safety Simulations
- Impact Analysis - Drop Tests, Bird strike, Penetrations, Explosives
- Metal Forming - Metal stamping, Hydroforming, Forging
- Failure Analysis

In addition, many times for such applications, instablities which are not clearly associated with a change in parameters occur and result in a stocastic variation of the response, compare *Roux et al.* [4]. This stochastic variation is usually induced by physical (e.g. bifurcations sensitive to initial values) and non-physical (numerical) effects.

In order to accomplish optimization for highly nonlinear applications in the following four different approaches are discussed.

2.1 Gradient-Based Optimization

For gradient based optimization the computation of sensitvities (gradients) is required. This can be done either analytically or numerically.

Analytical gradients with respect to design variables are formulated explicitly and implemented into the simulation code. But this is rather complicated and for highly nonlinear computations almost impossible.

Numerical gradients are obtained by the perturbation of n design parameters and $(n+1)$ simulations are performed. This is simple, but it is difficult to find a proper perturbation interval. If the interval is too large, there is a loss of accuracy, if it is too small spurious derivatives might be computed due to chaotic structural behaviour (noise) and numerical round-off errors.

The most popular gradient-based methods are SQP (Sequential Quadratic Programming) and MMA (Method of Moving Asymptotes, *Svanberg* [12]). Various tests and comparison of several gradient-based optimizaton methods are described in *Schittkowski* [7].

2.2 Random Search

Usually a sequential search method is applied by which the best design is selected from each iteration. A sorting procedure is used to select the design with the lowest (for minimization) or highest (for maximization) objective from all the feasible designs. If no feasible design exists, the least infeasible design is chosen. An experimental design such as Latin Hypercube Sampling (LHS) allows a sequential random search procedure. It is a common approach to automatically move the region of interest by centering it on the most recent best design.

Random search approaches usually show a rather poor convergence behaviour. The main reason for its popularity is probably its simplicity.

2.3 Genetic Algorithm

More intelligent search methods are genetic or evolutionary algorithms.

A typical scheme for a genetic algorithm (GA) is as follows:

1. Initially several individuals are randomly generated to form the first initial population $\boldsymbol{X}_s$. A single individual consists of a vector of design parameters.
2. Evaluate the individual fitnesses of the population. This means, computation of simulation results for each single individual.
3. Random selection of individuals of the population $\boldsymbol{X}_s$. Observations with a better fitness get a higher probability for the selection.
4. Generate a second population $\boldsymbol{X}_m$, based on mutation and recombination (cross-over)
   - *Mutation:* $\boldsymbol{X}_m = \boldsymbol{X}_s + \Delta\boldsymbol{X}$. The variation $\Delta\boldsymbol{X}$ is evaluated using a normal distribution with a mean of $\Delta\boldsymbol{X} = 0$ and a specified standard deviation σ_m. Thus, small changes occur often, large changes are generated rarely.
   - *Recombination:* Several individuals are mixed and recombined to new populations. Typically by simply swapping a portion of the data structure (design parameter values).
5. This generational process is repeated (continue at step 2.) until a termination condition has been reached. A common termination criteria is when the change of design parameters falls below a predefined value.

GAs can rapidly locate good solutions, even for difficult search spaces and GAs may have a tendency to converge towards local optima rather than the global optimum of the problem.

It is worth tuning the parameters of the genetic algorithm such as mutation probability, recombination probability and population size to find reasonable settings for the problem class that is considered.
For example, a very small mutation rate may lead to genetic drift or premature convergence of the genetic algorithm in a local optimum. A mutation rate that is too high may lead to loss of good solutions. Mutation rate is the ratio of the applied standard deviation σ_m for the perturbation $\Delta\boldsymbol{X}$ of iteration n and iteration $n+1$. *Self-adapted schemes* adjust the settings of the GA during the optimization on the basis of specific criterias.

For more details on genetic algorithms and evolutionary strategies the book of *Schwefel* [8] is highly recommended.

2.4 Optimization using Response Surfaces

Among several methodologies available to address optimization in a design environment, the Response Surface Methodology (RSM) has achieved prominence in recent years. The RSM is a statistical method for constructing smooth approximations to the objective function in the multi-dimensional parameter space. So called experimental design points (variable sets $\boldsymbol{x}_p$; $1, \ldots, p$; $p =$ number of experimental points) are selected within a pre-defined design space (Figure 1). For the parameter combinations of these experimental points Finite-Element simulations are performed and specified model responses are calculated. In a subsequent step polynomial functions are fitted to these experimental design points in order to substitute the original, possibly very noisy, response.
For illustration purposes: In the two-dimensional parameter space ($n = 2$) the polynomial functions represent surfaces, which are adapted to selected experimental points within the three-dimensional space, see Figure 2. The fit of the polynomial functions is done by using regression analysis. Least squares approximations are commonly used for this purpose.

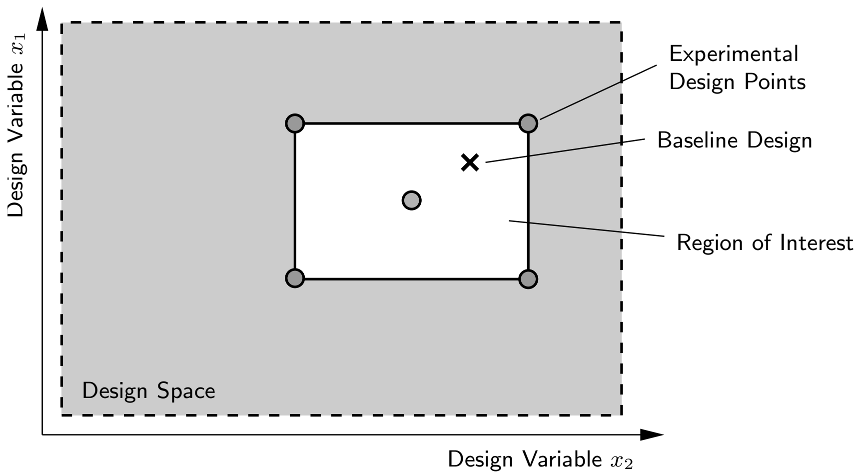

Figure 1: Design space, region of interest and experimental design points. The design space is a strict bound for the design variables, the region of interest is the currently considered domain and the experimental design points are evaluated by a specified type of *Design of Experiments*, compare Section 2.4.1

2.4.1 Design of Experiments

Experimental Design is the selection procedure for finding the points in the parameter design space. Many different methods are available, e. g. Koshal Design, Factorial Design, Central Composite Design, Box-Behnken Design, D-Optimal Design etc. An excellent review of the different design types can be found in *Myers and Montgomery* [3].
An advantage of the D-Optimal Design is, that design regions of irregular shape and any number of experimental points can be considered. The experimental points for the D-Optimal Design are usually selected from a full factorial design by using the D-optimality criterion. The number of experimental points is of course in correlation with the order of the approximation functions. Usually oversampling of approximately 50 % is recommended (*Roux et al.* [5]), i. e. 50 % more points are being analyzed than the minimum required.

2.4.2 Approximations

Polynomial functions might be composed by arbitrary base functions. The selection of the base functions should lead to a best regression model. The use of full quadratic approximations (second-order model) is very common, but because of their cost they should be avoided for very large models. A possible solution is to use linear approximations. These are generally inaccurate beyond the immediate neighborhood of the considered design point, but can be used in a successive response surface procedure.

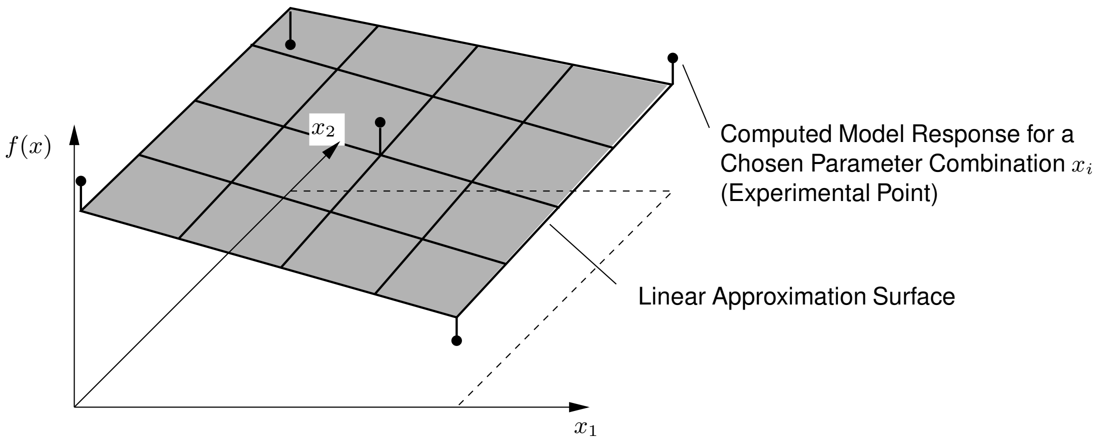

Figure 2: Approximation surface is fitted through points in the design space

2.4.3 Successive Response Surface Method - SRSM

For the successive response surface method a region of interest is defined as a sub-region of the entire design space (Figure 1). The sub-region is approximated and the optimum is determined on the approximated response surface. Then a new region of interest is defined and the center is located on the previous successive optimum. Progress is made by moving the center of the region of interest as well as reducing its size (*Stander* [10]), compare Figure 3. The iterations are continued until the objective function or the design variables reach stationary values.

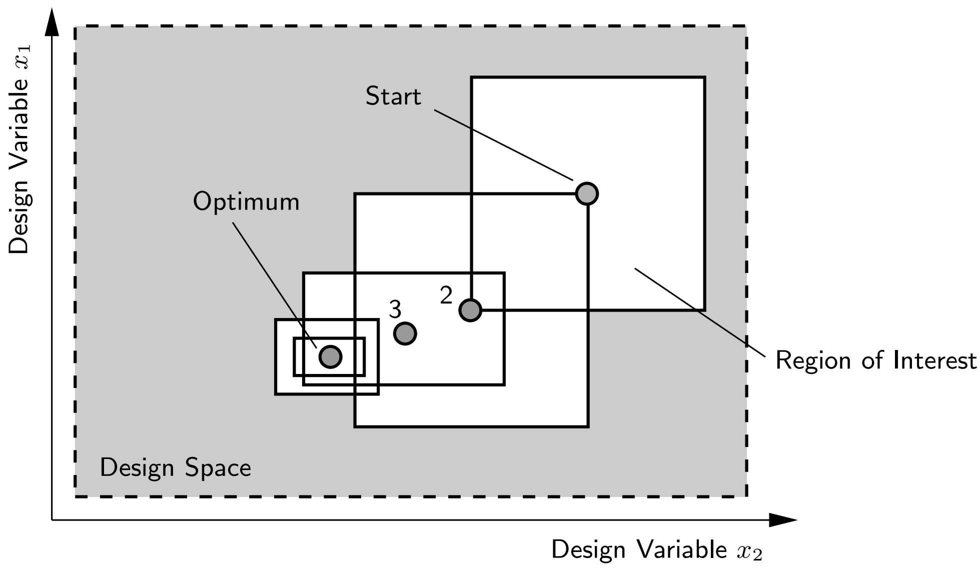

Figure 3: Application of the successive response surface method (SRSM) in LS-OPT. Each subregion represents one iteration. As response surfaces for the subregions linear, quadratic or even neural networks can be applied.

The successive sub-problems of the SRSM method are solved using a multi-start variant of the leap frog dynamic trajectory method, LFOPC (*Snyman* [9]). It is a gradient based method, constraints are incorporated by a penalty based function.

2.4.4 Global Approximations using Neural Networks

Besides polynomial response surfaces there are several other metamodeling techniques, such as *Kriging*, *Neural Networks* or *Moving Least Square Methods*. Neural networks have substantial advantages and thus will be surveyed in the following.
Linear polynomials are highly suitable for optimization using the successive response surface scheme and for variable screening using ANOVA, see *Craig et al.* [2]. For a more accurate approximation covering a larger range of the design space (global approximation) nonlinear approximations have to be considered. Higher order polynomials tend to show oscillations between the simulation points and at the boundaries of the design space, while neural networks are able to capture strong non-linearities without such effects. At the core, polynomial approaches and neural nets differ in the regression methods that they employ to construct the surrogate models. The polynomial response surface method uses linear regression, while neural networks use nonlinear regression methods, which require optimization algorithms such as Quasi-Newton methods (BFGS) or the Levenberg-Marquardt algorithm, see *Bishop* [1]. The evaluation of the regression coefficient is usually refered to the term *network training*. For the distribution of the sampling points a *Space Filling DOE* or a *Latin Hypercube Sampling* is recommended.

Feed-forward (FF) networks are the default topology for neural networks in LS-OPT. The restriction for FF-networks is that the topology diagram must be feed-forward, so that it contains no feedback loops. This ensures that the network outputs can be calculated as explicit functions of the inputs and the weights. In Figure 4 a diagram of a two-layered feed-forward network is displayed. Units which are not treated as output units are called hidden units.

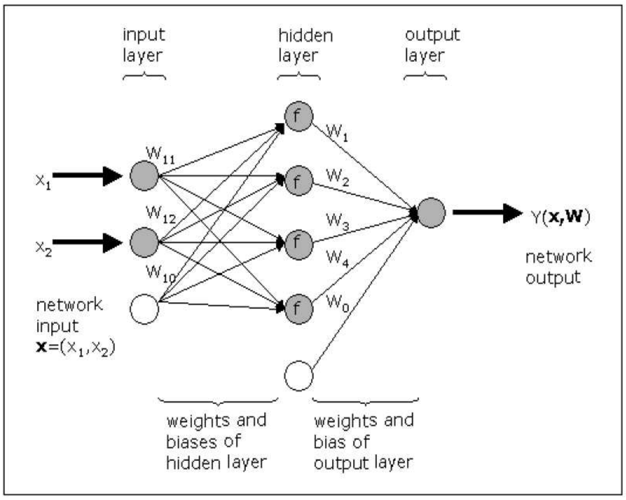

Figure 4: Schematic of a neural network with 2 inputs and a hidden layer of 4 units with activation function f

The output of each layer of units are the inputs to the next layer. In a feed-forward network, the *activation function* of intermediate (hidden) layers is generally a *sigmoidal function* (Figure 5), network input and output layers being linear. The term sigmoid means *S-shaped*.
Consider a FF network with K inputs, one hidden layer with H sigmoid units and a linear output unit. For a given input vector $\boldsymbol{x} = (x_1, \ldots, x_k)$ and network weights $\boldsymbol{w} = (w_0, w_1, \ldots, w_H, w_{10}, w_{11}, \ldots, w_{HK})$, the

output of the network is

$$\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{h=1}^{H} w_h \, f\Bigg(\underbrace{w_{h0} + \sum_{k=1}^{K} w_{hk} x_k}_{a}\Bigg), \tag{1}$$

where

$$f(a) = \frac{1}{1+e^{-a}} \quad . \tag{2}$$

Equation (1) demonstrates how the output of the linear input layer becomes an input for the sigmoidal function of the hidden layer. The extension to the case of more than one hidden layer can be obtained accordingly. In LS-OPT one hidden layer is used as displayed in Figure 4. The number of hidden (sigmoid) units H varies (automatically) within 0 and 10, dependent on the curvature (nonlinearity) of the problem and on the number of available simulation points.

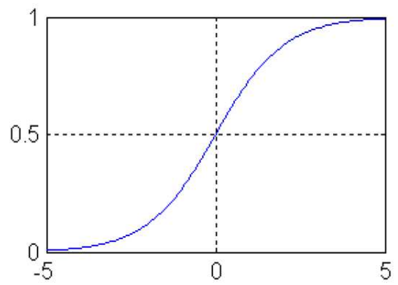

Figure 5: Sigmoid transfer function $y = 1/(1+e^a)$ typically used with feed-forward networks

More details regarding the implementation of neural networks in LS-OPT can be found in the LS-OPT manual *Stander* [11]. In this manual, topics such as *Regularization*, *Error Indicators*, *Back Propagation* and the *Variability* of feed-forward neural networks are discussed.

An excellent review of pattern recognition using neural networks is given in *Bishop* [1].

The advantages of neural nets for nonlinear regression might be summarized as follows:

- Global approximation
  - Can apply design exploration or supplementary studies after optimization
  - Allow local refinement in regions where many simulation points are available, but maintain global relevance
- Do not have to choose different orders of polynomials
  - Choice of neural network architectures is automated
  - Independent of number of experimental points, even a minimum, e.g. as for linear polynomials, can be used. In contrast to different polynomial orders, which require a discrete number of runs (e.g. 10 Variables: for linear at least 11 points, for quadratic at least 66 points)
- It is a regression method, not an interpolation method
  - Smoothing and averaging of responses
  - filter out noise

2.5 Summary of Optimization Strategies

In Table 1 pro and cons of several optimization strategies for highly non-linear problems are listed.

	Gradient Based	Random Search	Genetic Algorithm	RSM / SRSM
⊕	▷ number of solver calls ▷ accuracy of solution	▷ very robust, can not diverge ▷ easy to apply	▷ good for problems with many local minimas	▷ very effective, particularly SRSM ▷ design space exploration using Response Surfaces ▷ filter out noise, smoothing of results
⊖	▷ step-size dilemma for numerical gradients ▷ can stuck in local minima ▷ can diverge	▷ bad convergence, not effective ▷ chooses best observation - may not be representative of a good (robust design)	▷ many solver calls, only suitable for fast solver runs ▷ chooses best observation - may not be representative of a good (robust design)	▷ approximation error ▷ verification run might be infeasible

Table 1: Overview of different optimization strategies for highly nonlinear applications

3 Optimization of an Adaptive Restraint System for Several Front Crash Load Cases

3.1 Description of Load Cases and FE-model

3.1.1 Load Cases

The ideal restraint system decelerates the occupant as fast as possible on a constant acceleration level. Different masses of the occupants and thus different load cases mean for an ideal deceleration behaviour, that the restraint system must be adapted to the required force levels ($F = m\,a$). With the today's state of technology it is possible to identify the load case and the different types of occupants. This means, system parameters of the restraint system, such as trigger time for seat-belt, airbag and steering column might be adapted to specific load cases.

For the optimization problem presented in this Section, 4 different front-crash load cases (FMVSS 208) are taken into consideration:

- *H305a:* Hybrid III 5th female dummy; 56km/h - belted
- *H305p:* Hybrid III 5th female dummy; 40km/h - **not** belted
- *H350a:* Hybrid III 50th male dummy; 56km/h - belted
- *H350p:* Hybrid III 50th male dummy; 40km/h - **not** belted

3.1.2 Finite-Element Models

The finite element PAM-crash models for the four considered load cases have the following properties:

- about 500000 elements

- FE-models for Hybrid-III 5th female and Hybrid III 50th dummy male from FTSS [1]
- wall clock simulation time approx. 19h on 4 cpus, distributed memory

In Figure 6 the FE-Model with a belted Hybrid III 50th male dummy is shown. It represents the load case *H350a*.

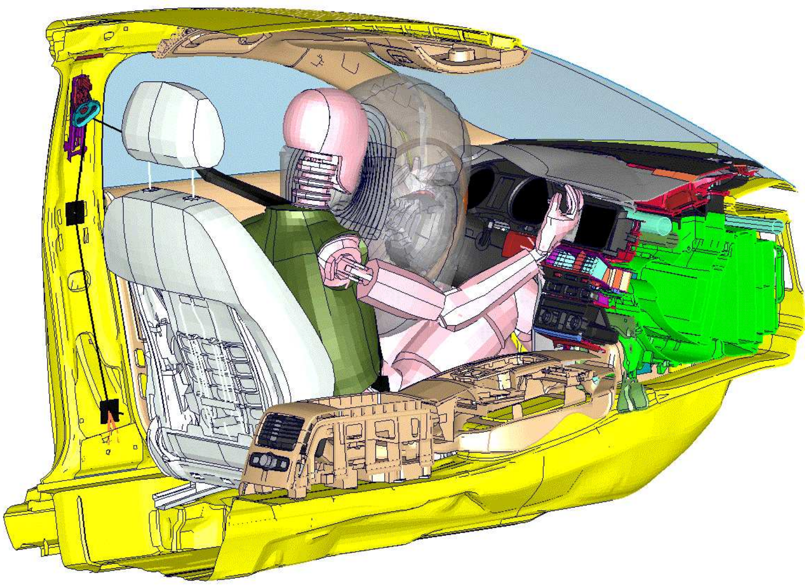

Figure 6: Finite-Element crash model with a belted Hybrid III 50th male dummy (cross-section)

3.2 Multi-Disciplinary Optimization Problem

Goal of the optimization is to adapt the adjustments of the restraint system in order to optimize the occupant safety performance for all four loadcases H305a, H305p, H350a and H350p simultaneously.

3.2.1 Design Variables

Some system parameters as for example time to fire (TTF) of the airbag might be set individually for each load case. Due to different identification technologies the restraint system can recognize a specific loadcase and assign an associated TTF-value.
Other system parameters such as venthole diameter of the airbag can of course not be adapted individually to the different load cases. Thus, these parameters have to be set globally. After each iteration of the SRSM, in LS-OPT the variables are updated to ensure a unique intermediate design for the multiple disciplines (load cases). In Figure 7 an FE-model of the restraint system is shown.
In Table 2 all the parameters of the restraint system are summarized, which might be considered as design variables for the optimization.

3.2.2 Objective

The objective of the optimization is to minimize the thorax acceleration in terms of the *a3ms*-criteria described in [6]. This is applied to all four solver cases with respect to a muli-objective function with equal weights, see Section 3.2.4.

[1] First Technology Safety Systems

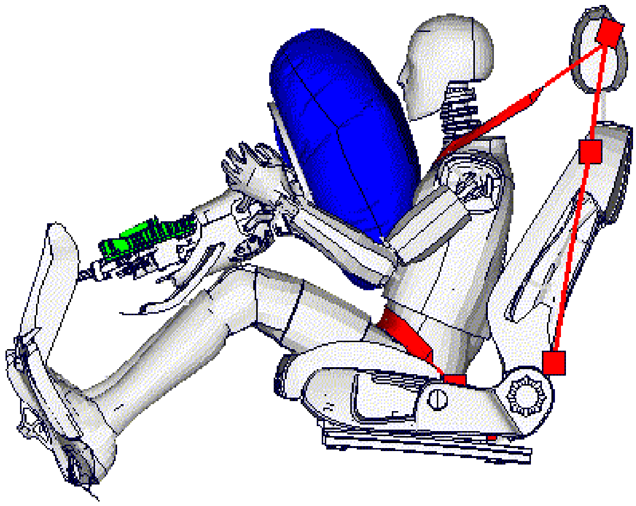

Figure 7: Restraint system with adaptive components for airbag, steering wheel and seat belt

Adaptive System	Design Variable	#	Load Case
seat belt	Upper force level	1	H305a/H350a
	Trigger time female belted	2	H305a
	Trigger time male belted	3	H350a
airbag	Area initial vent hole	4	H305a/H305p/H350a/H350p
	Area additional vent hole	5	H305a/H305p/H350a/H350p
	Trigger time female belted	6	H305a
	Trigger time female unbelted	7	H305p
	Trigger time male belted	8	H350a
	Trigger time male unbelted	9	H350p
steering wheel	Lower Force Level	10	H305a/H305p/H350a/H350p
	Trigger time female belted	11	H305a
	Trigger time female unbelted	12	H305p
	Trigger time male belted	13	H350a
	Trigger time male unbelted	14	H350p

Table 2: List of design variables for the respective load cases

3.2.3 Constraints

Four different, typical dummy responses for each load case are taken into consideration:

- HIC15 - **H**ead **I**njury **C**riteria for 15ms, evaluation see [6]

- Femur Forces (left/right)
- Thorax Acceleration
- Thorax Intrusion

These responses are considered as constraints in order not to exceed 80% of the maximal value required by regulations.

3.2.4 Approach of Optimization

Optimization is performed in two steps.

First Step: Minimize the Maximum Constraint Violation (MinMax-Problem)

$$\begin{aligned} &\text{Minimize} && e \\ &\text{subjected to} && g_i(\boldsymbol{x}) \leq U_i + e \\ & && e \geq 0 \end{aligned} \tag{3}$$

With e as an additional auxiliary variable and $g_i(\boldsymbol{x})$ as the constraints listed in Section 3.2.3 with the according upper bounds U_i. For $e = 0$ there is no constraint violation and thus the design is feasible.

Second Step: Minimize thorax accelerations (multi-objective)

$$\begin{aligned} \min f(\boldsymbol{x}) \quad = \quad & \text{a3ms}_{thorax-accel}(\text{H305a}) + \text{a3ms}_{thorax-accel}(\text{H305p}) + \\ & \text{a3ms}_{thorax-accel}(\text{H350a}) + \text{a3ms}_{thorax-accel}(\text{H350p}) \end{aligned} \tag{4}$$

3.3 Process Flow for Optimization

To apply the above described optimization an automated process flow for the crash simulations has to be carried out. In Figure 8 the procedure is displayed in a flow chart. Firstly, the preprocessing of the input files is performed, by the substitution of the process variables, which are evaluated by LS-OPT. Then the jobs for the several load cases are submitted to a computing cluster via LSF. While the jobs are running on the cluster LS-OPT is monitoring the progress of the runs. After the jobs are finished the results are extracted by the software EVALUATOR [2] and be transfered to LS-OPT. Evaluation and interpretation of the results in LS-OPT leads to a new set of process variables (design variables) unless the optimization is converged.

The challenging task for the automated process flow is to get it as stable as possible over a long period of time. Many software components and several hardware platforms and disks are involved, which can easily lead to a fragile system.

3.4 Results

As the methodology for the optimization the successive response surface method (SRSM) is applied, which is described in Section 2.4.3. Since the simulation time of a single PAM-Crash run is rather time consuming, fast convergence of the algorithm is important.
In order to achieve for design exploration a response surface with a reasonable global approximation, simulation points in sparse regions are added and a neural network is fitted to all points.

[2] Product of GNS mbH, Braunschweig, Germany

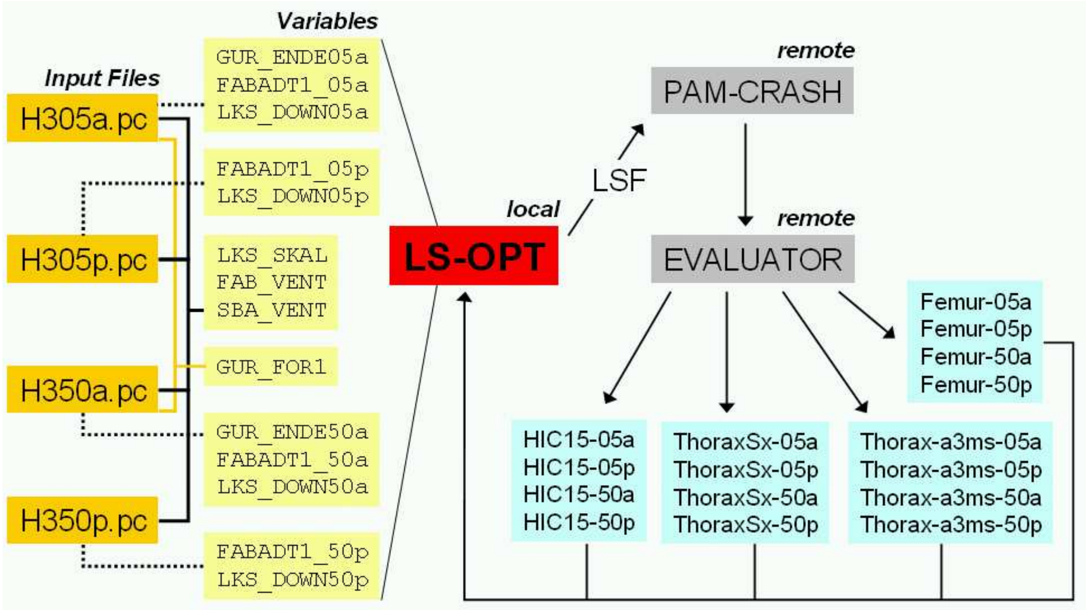

Figure 8: Process flow for optimization

3.4.1 Optimization using SRSM with Linear Polynomials

In the baseline design the constraints listed in Section 3.2.3 are heavily violated. The main goal of the optimization is to find a feasible design, which satisfies the constraints of all four load cases. The algorithm in LS-OPT solves an internally auxiliary problem by minimizing the variable e, see equation (3).
In Figure 9 (left) the optimization history of the constraint violation is shown. The values of the max constraint violations for the eight iterations do not always refer to the same constraint criteria. Small changes of the parameter values can lead to a significant change in the responses and thus swap the maximum violated constraint. This is for example the case when the airbag is too soft and the head of the dummy strike through onto the steering wheel. In this case, the HIC15 response is almost like a discontinous function. This effect can be seen in iteration 4 and 6 in the optimization history plot of the maximum constraint violation (Figure 9, left), where the HIC15 value cause a very large maximum constraint violation. Such effects are a challenging task for an optimization algorithm. In the 8^{th} iteration the constraint violation drops down to zero and minimization of the multi-objective function (4) in Section 3.2.4 is performed subsequently.
For the displayed optimization results in Figure 9 adaptive variables for the steering wheel (Table 2) are not considered. This means, in total there are only 9 design variables for the four load cases, 5 variables for the active load cases H305a and H350a, 2 variables for the passive load cases H305p and H350p and 2 global variables for all load cases.
In total 272 crash simulations for all four load cases are performed within the 8 iterations. Within previous studies for the same problem an evolutionary algorithm has been tested. For this, approx. 30-40% more simulations had been necessary to achieve a similar result.

3.4.2 Neural Network Surface for Global Design Space Exploration

For the successive response surface scheme with linear polynomials only for the first iteration a global approximation might be performed. This approximation is built with very few points and is usually a very rough guess of the global system behaviour. During the successive scheme the considered region becomes smaller

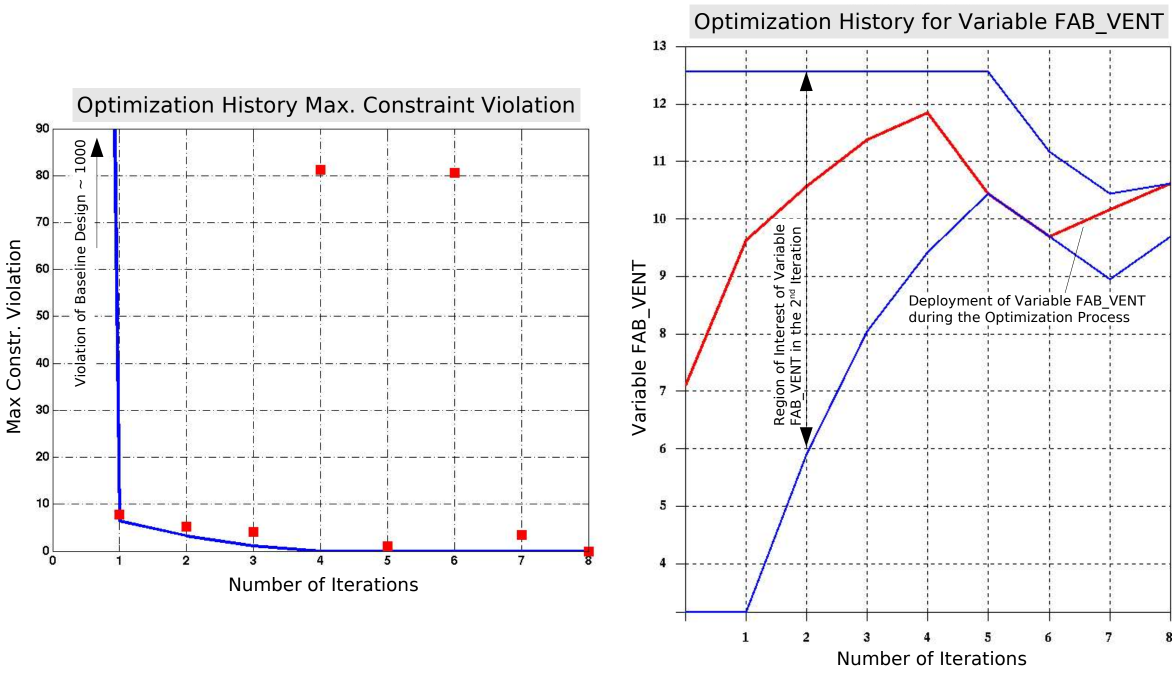

Figure 9: *Left:* Optimization history of maximum constraint violation. In Iteration 0 (baseline design) the violation is approx. 1000, after 8 iterations the maximum constraint violation is equal to zero. This means, all the constraints listed in 3.2.3 are for each load case fulfilled. *Right:* As an example the deployment of the variable FAB-VENT is shown during the successive response surface scheme.

and thus experimental points are sampled only within this mid-range or local regions, compare Figure 3. This leads to a non-uniform distribution of simulation points through the design space. This is not ideal in order to get a good and reliable global approximation by a non-linear response surface, such as neural nets (see Section 2.4.4).
Therefore, LS-OPT provides the capability of adding experimental points using a space filling algorithm, see *Stander et al.* [11]. In Figure 10 it is shown how 50 points using the space filling method are added to all the points achieved during the successive scheme in each iteration. This is performed to distribute additional experimental points as equally spaced as possible in order to get consistently simulation points in the whole design space. Thus, a reliable neural net for a global approximation can be established. The space filling algorithm uses *simulated annealing* in order to maximize the distances between the additional experimental points and the existing points.
In can be seen, that in the region where the minimum of the surface is located, a very high density of points occur. This is due to the zooming of the region of interest towards the minimum of the response surface, applied by the successive response surface scheme.

3.4.3 Visualization of Response Surfaces

In order to explore the design space and to evaluate relationships between variables and responses the software D-SPEX is used. D-SPEX allows the visualization of 2-dimensional curve plots as well as 3-dimensional surface plots. For an $n > 2$-dimensional ($n =$ number of design variables) problem two (or one) variable can be selected while the other variables values can be varied by a slider. In addition, viewing of simulation points can be added in order to account for the residuals and the variance with respect to the response surface, compare Figure 12. This indicates a measurement of the noise of the problem, *Roux et al.* [4].

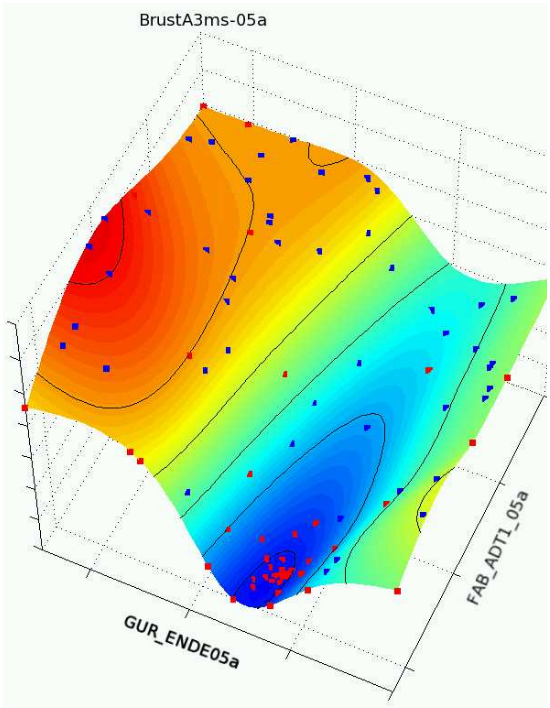

Figure 10: 50 additional points are added using a *Space Filling Design of Experiments* in order to get a reliable global approximation with a Neural Network Surface. Visualized simulation points are projected onto the response surface.

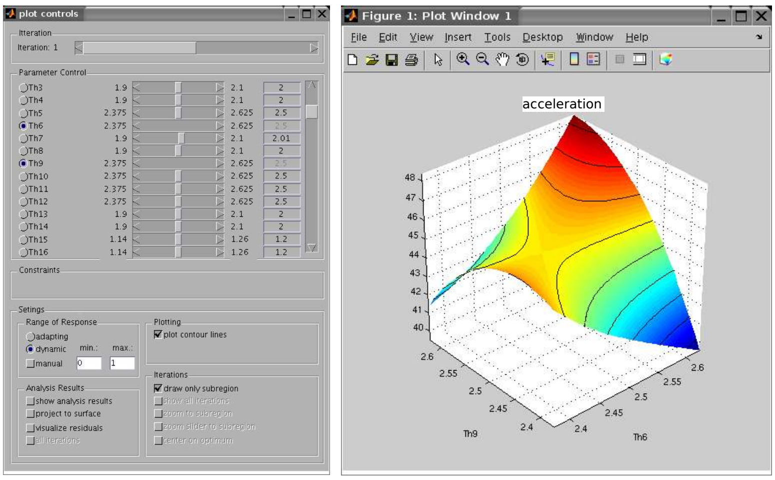

Figure 11: Control panel of D-SPEX and 3D-plot of a neural network. The sliders allow to vary parameters of the multi-dimensional problem which are not displayed in the plot window.

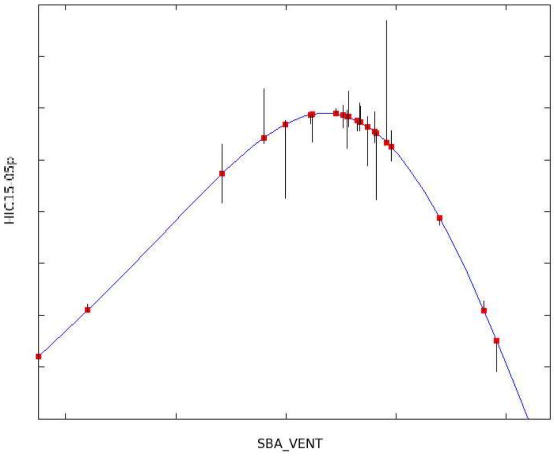

Figure 12: 2D-plot of a multi-dimensional problem. Simulation points are projected onto the curve. The vertical bars indicate the residuals of the simulation points with respect to the response surface.

4 Conclusions

In this paper the successive response surface scheme has been applied successfully to the optimization of an adaptive restraint system considering several front crash load cases at AUDI.

Starting from a highly infeasible design after eight iterations a feasible solution was established by the meaning of satisfying all constraints for all load cases. With the optimization a combination of parameters for the adaptive restraint system has been found, which results for the FE-simulations in response values significantly lower than the pre-defined requirements. In total nine design variables have been considered. However, not all variables are used in each load case, some of them are fully shared and some are partially shared. For this type of optimization problem the SRSM seems to be a suitable and effective methodology.

Among system optimization it might be useful to have a mathematical approximation model (*Meta-Model*) in order to explore relationships between variables and responses. This has been performed for the considered restraint system problem using a feed-forward neural network.

Further investigations are put into the direction of stripping down the PAM-crash model for even more comprehensive optimization tasks. It seems to be possible to achieve simulation times of about 4 hours per run with almost similar results as for the full-scale problem. Thus, substantially more load cases with even more design parameters could be optimized simultaneously within an acceptable response time.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York, 1995.

[2] K. J. Craig, N. Stander, D. A. Dodge, and S. Varadappa. Automotive crashworthiness design using response surface-based variable screening and optimization. *Int. J. Comp.-Aided Engng and Sofw*, 22:38–61, 2005.

[3] R. H. Myers and D. C. Montgomery. *Response Surface Methodology - Process and Product Optimization Using Designed Experiments*. J. Wiley, New York, 1995.

[4] W. J. Roux, N. Stander, F. Günther, and H. Müllerschön. Stochastic Analysis of Highly Nonlinear Structures. *Int. J. Numer. Meth. Engng*, accepted July 2005.

[5] W. J. Roux, N. Stander, and R. T. Haftka. Response Surface Approximation for Structural Optimization. *Int. J. Numer. Meth. Engng*, 42:517–534, 1998.

[6] Working Group Measured Data Processing Vehicle Safety. Crash Analysis - Criteria Description. Version 1.6.2, 2005.

[7] K. Schittkowski, C. Zillober, and R. Zotemantel. Numerical Comparison of Nonlinear Programming Algorithms for Structural Optimization. *Strutural Optimization*, 7:1–28, 1994.

[8] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley and Sons, New York, 1995.

[9] J. A. Snyman. The LFOPC leap-frog algorithm for constrained optimization . *Computers and Math. with Applic*, 40:1085–96, 2000.

[10] N. Stander and K. J. Craig. On the robustness of a simple domain reduction scheme for simulation-based optimization. *19*, 4:431–450, 2002.

[11] N. Stander, T. Egglestone, K. Craig, and W. Roux. *LS-OPT User's Manual - Version 2*. Livermore Software Technology Corporation, Livermore, 2004.

[12] K. Svanberg. MMA - Method of Moving Asymptotes - A New Method for Structural Optimization. *Int. J. Numer. Meth. Engng*, 24:359–373, 1987.

