Multidisciplinary Optimization considering Crash and NVH Loadcases

Andreas Hoppe, Michael Kaufmann
Audi AG, Ingolstadt, Germany

Boris Lauber
FE-DESIGN GmbH, Karlsruhe, Germany

1. Introduction

The development of a car body is a very complex process, as various functional requirements have to be considered. Different targets with sometimes competitive issues define a challenge for the development engineers. Today, numerical simulation is a well established and widespread tool to predict, analyse and optimize the functional characteristics of a car. The engineer describes a design and the simulation will predict the performance of this design. Very often, numerical optimization is performed considering only one single simulation discipline. Once an optimal design has been found it has to be verified for all other relevant disciplines. In most cases this process leads to significant design changes. Therefore, it requires a great effort of all participating engineers to find an optimum design that satisfies all relevant disciplines. This process is very time consuming and often ends up in an unsatisfying design compromise.

The aim of the multidisciplinary optimization (MDO) approach is to consider all participating disciplines that are defining constraints on the design in one single numerical optimization procedure. Very high computational resources are required to solve the problem in a reasonable amount of time, as multiple loadcases from different analysis disciplines with different software products have to be considered simultaneously. In the last few years, computer hardware has rapidly increased in performance. Therefore the calculation of the many necessary design variants for an MDO can now be carried out in a time span short enough to be considered as practicable

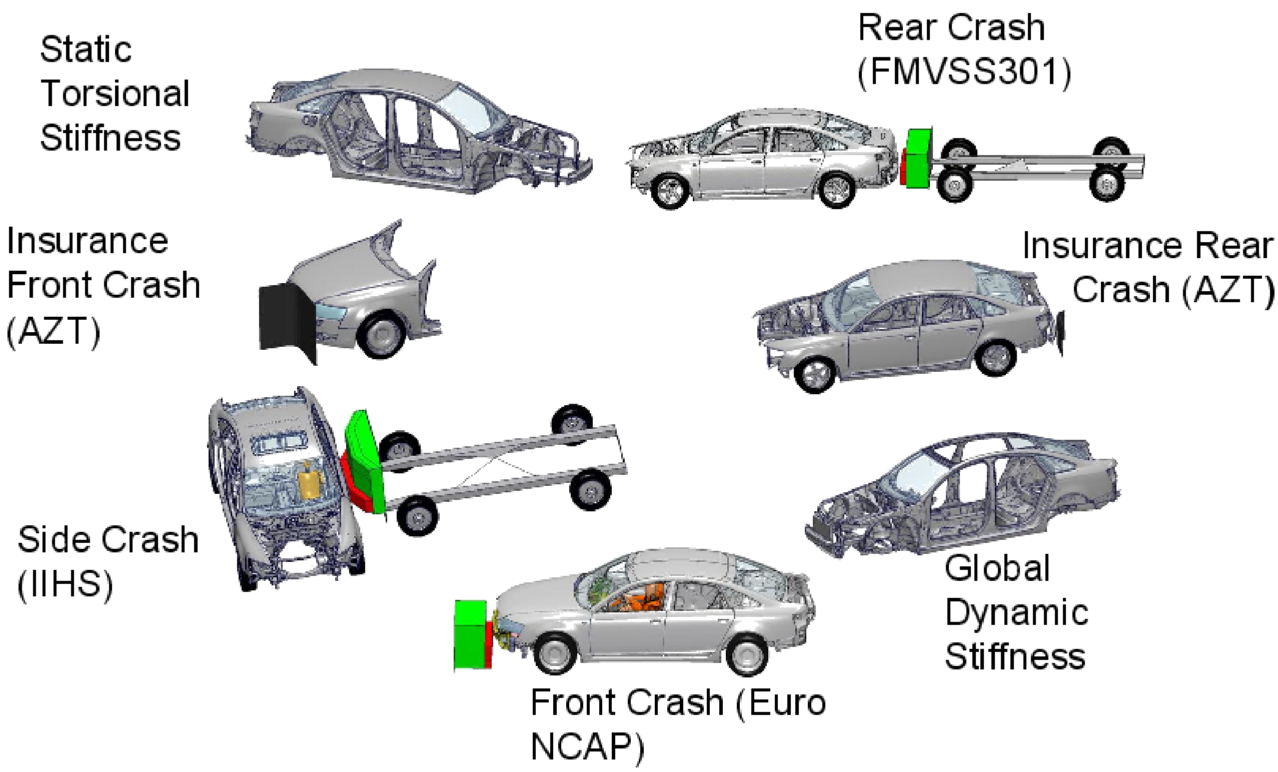

Fig. 1: Considered disciplines in full vehicle MDO

within a modern car development process.

In addition to the turnaround time of the MDO itself, one has to consider the time needed for the setup and the result evaluation of the MDO process. Therefore, an integration platform for the setup, the execution and the exploration of a multidisciplinary optimization has to be provided. OPTIMUS is a software tool that helps the engineer to define and automate existing simulation workflows independent on the applied simulation software. Once the simulation process is captured, it may be used for different analysis purposes. Design of Experiments (DOE) methodologies may be used for a screening of the design space or for the generation of approximation models using Response Surface Modelling (RSM) techniques. Additionally, classical optimization methods are available. The global and local optimization techniques may be executed on the original simulation workflows or on the approximate models. They allow one to solve general non-linear programming both for single or multi objective problems with arbitrary constraints. Several resource management systems can be applied for the parallel execution of the virtual experiments.

2. Multidisciplinary Optimization

Many requirements have to be checked carefully before one can start a multidisciplinary optimization as they have a large influence on the appropriate strategy.

- Problem size from component level to complete systems
- Number of loadcases and analysis disciplines
- Existing time frame and IT resources
- Number and type of design variables
- Number of objectives and constraints
- Types of analysis (linear, non-linear)
- Optimization in the early concept phase or design improvements

For every discipline to be considered in the MDO, the simulation workflow has to be automated in order to be executed multiple times. The automation must include the entire workflow from preprocessing respectively the setup of the analysis inputdeck to the postprocessing. The user has to assure that all necessary results needed for a complete design evaluation as a basis for a successful optimization are extracted. If the maximum intrusion of a certain point in a crash analysis of a car structure has to be limited in an optimization, the user has to automate the generation of the corresponding displacement curves that have to be provided to the optimization system for the interpretation of the design. Sometimes, objectives which seem very simple in the interactive evaluation of a design are very complicated to automate. For example the necessary space for the side-airbag between seat and door during a side-crash is more than just one value and this visual criterion has also to be defined to allow the optimization to rate the results.

2.1. The Optimization System

The optimization system plays a key role in the execution of multidisciplinary optimization. The driving system for a multidisciplinary optimization needs to fulfil certain requirements:

- Integration of any simulation tool
- Efficient capturing of analysis workflows
- Possibility of doing variant calculations
- Different optimization techniques
- Postprocessing capabilities

- Capabilities for parallelization of the virtual experiments

2.1.1. Process Automation

First, the simulation workflow on which the MDO is based upon has to be described. All design variables (shell thicknesses, material properties, parameters, …) have to be provided as input values. The input values have to be replaced in the simulation input files. This is realized by template techniques. The analysis has to be executed and the results have to be generated from the analysis results files. Finally, output values have to be extracted for which the objective and constraints can be defined. The methodology enables the engineer to set up the sequence of analysis tools graphically. The analysis sequence specifies the complete flow of information – all participating disciplines may be described in parallel workflow branches depending on the same input parameters.

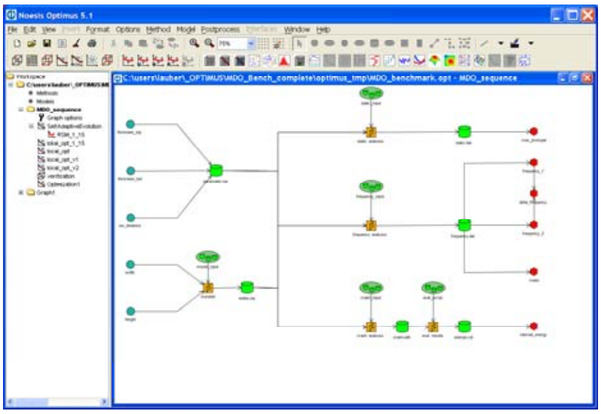

Fig. 2: Process automation in OPTIMUS

Once the workflow is described, it can be executed many times. This allows the automated evaluation of variants and therefore allows the optimization system to find the optimal configuration of the inputs to improve the objective including all the constraints.

2.1.2. Design Optimization

For an MDO an efficient optimization approach has to be selected in order to improve the design with as low computational effort as possible. For MDO problems including crash simulation we will find a highly non-linear behaviour of the objective function. Therefore, the use of local gradient based optimization algorithms is not suitable. Local search algorithms strongly depend on the selected start value. Thereby they have a high probability to be trapped in a local minimum and have a poor performance due to sequential function evaluations.

Evolutionary algorithms are more efficient for MDO projects due to the following reasons

- Evolutionary algorithms have a lower risk to be trapped in a local minimum. In the starting population a good coverage of the design space is obtained by random sampling of the input parameters.
- Parallelization plays a very important role in the use of evolutionary algorithms. Each member of a population is completely independent of the other population members. This allows executing the analysis for all members of one population in parallel.
- Evolutionary algorithms are very robust. If one ore more results for a calculated design are missing, the stability of the algorithm will not be affected.

2.1.3. Efficient and robust process workflow

OPTIMUS supports distributed and parallel computation – independent experiments may be spread on a computation cluster to shorten the overall optimization time. A certain number of experiments may be executed in parallel dependent on the available computer resources. Therefore a certain number of generations may be evaluated. OPTIMUS is using fine grain parallelizing techniques. It first deducts a task schedule by breaking down the analysis sequence to tasks that need to be performed and determines dependencies among these tasks. Then it assigns each task to the available CPU. This method allows for scheduling computationally expensive tasks to high performance nodes, while submitting lower level tasks to the rest. It also allows for assignment of tasks to nodes on the cluster on which a particular simulation software is licensed without loosing the ability to submit other tasks in more nodes of the cluster. In combination with the restart capabilities for the global algorithms the complete optimization workflow can be designed as robust as possible to avoid the loss of results during the optimization procedure.

2.1.4. Postprocessing using Response Surface Modelling Techniques

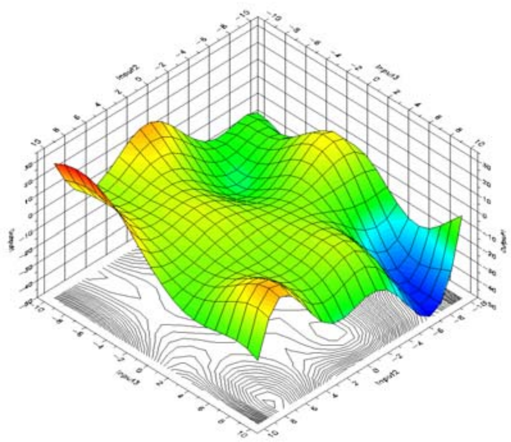

Fig. 3: Stochastic interpolation RSM

Global optimization algorithms have a high probability to improve the existing design. All the knowledge, that is generated during the global optimization should be used to improve the design. The data may be used for the generation of a surrogate model. This surrogate model may then be used for additional optimization runs in the design space covered by the preceding global optimization algorithm. The function evaluation is very fast as it is made on the approximated analytical model and enables the user to perform quick interactive optimizations.

Finally, the improvements found on the basis of the surrogate model have to be validated on the original simulation sequence to check if all the constraints are fulfilled.

The representative types of available RSMs are polynomial type RSMs with and without stochastic correction terms. The general form of the RSM is:

$$RSM(\mathbf{x}) = \sum_{i=1}^{n} a_i * F_i(\mathbf{x}) + Z(\mathbf{x})$$

where i represents the number of approximating functions, a_i are the coefficients to be determined through Least Squares, $F_i(\mathbf{x})$ are the polynomial -or any user defined- mathematical functions. In the case of pure Taylor Polynomials the $Z(\mathbf{x})$ is set to zero.

A second type of RSMs are based on Stochastic Interpolation. In that case $F_i(\mathbf{x})$ is considered to be a constant. In computer experiments, observations are made on a response function by running the analysis sequence. Since this is exact $Z(\mathbf{x})$ represents the systematic departure from the assumed model. Usually the form of $Z(\mathbf{x})$ is unknown. Our approach is to model $Z(\mathbf{x})$ as a realization of a stochastic process in which the covariance structure of $Z(\mathbf{x})$ relates to the smoothness of the response. $Z(\mathbf{x})$ is assumed to have zero mean and a covariance matrix given by:

$$\mathrm{cov}\left(Z(\mathbf{x}_1), Z(\mathbf{x}_2)\right) = \sigma^2 \, e^{-\theta\left\|\mathbf{x}_1-\mathbf{x}_2\right\|^2}$$

Scale Parameter

Non-linear Model Parameter

3. Examples for MDO Applications at AUDI AG

During the last years the functional requirements of a car body have increased rapidly and this trend will continue. A car has to be very safe and must fulfil an enormous number of legal regulations, show a good performance in consumer tests concerning crash behaviour, should have a sporty dynamic chassis but must also be comfortable, quiet and light. Additionally, the different design requirements to fulfil crash load cases are not necessarily compatible with each other and at last, the costs play an important role. Unfortunately, these demands are often in conflict with each other and it is the task of the engineers to solve this complexity. Today, in some cases with many different functional requirements and constraints this complexity exceeds the level that can be handled without a systematic multidisciplinary optimization approach. At AUDI AG different projects using the optimization software OPTIMUS were performed and tested in their possible fields of application.

3.1. MDO of a Full Car considering Crash and NVH Loadcases

In one of the initial projects performed at AUDI AG using MDO the focus was not only the result of the optimization but more the required effort for defining the optimization and workflow, the stability of the process and the handling for postprocessing.

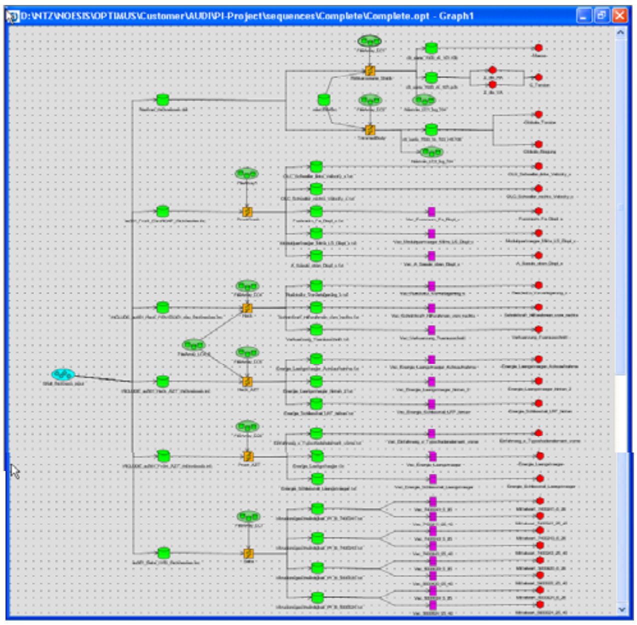

Fig. 4: Workflow for full vehicle MDO

The aim was to reduce the weight of a car body by changing 96 sheet metal thicknesses. Five crash load cases and two NVH loadcases with 28 constraints were considered. The optimization run should be realized within approximately two weeks. PamCrash and MSC.Nastran were used for solving. They were running on different compute clusters using different number of CPU's and had execution times between 18min (Static Torsion) and 22h (Frontcrash). PamView, In-House tools and scripts running on different platforms did the postprocessing of the 28 result values. The number of simulations was over 3500 and for all of them the process should run stable.

A process was defined and ready to start after a setup-phase of 8 days including the implementation of OPTIMUS into the AUDI IT-environment. It contained many steps to ensure stability of the process. The file-transfer was reduced by minimizing the number of files. Check-routines after copy-processes were implemented and loops for starting postprocessing and copying were added. On the basis of 800 CPUs available and the goal of 2 weeks execution time a Self Adaptive Evolutionary Algorithm with 100 designs per iteration and 5 iterations was selected. The two different clusters were loaded by OPTIMUS having several simulations running in parallel. During the process the long-term simulations (22h Frontcrash) were automatically preferred opposite to other simulations like Sidecrash (11h). This was necessary to guarantee a continuous load without blocking a significant amount of CPU's while waiting on one missing simulation.

Another aim of this project was to get an impression of how fast a significant reduction in weight can be achieved even if no information of the input parameters exists. In this case the existing shell thicknesses of the model were not used as initial design parameters. Therefore a large design space for each part was allowed. This results in an increasing amount of calculations but on the other hand this also increases the possibility to find an optimum. After the 5th iteration a considerable reduction in weight was achieved but the optimization was still not converged and there was still potential for further weight reduction.

The optimization progress was continued by building a surrogate model. After such a model is created an optimization can be executed on a PC in a few minutes. The proposed design was verified and achieved a weight reduction of several kg without violating any of the 28 constraints.

Now the experiences of this project with a more or less methodological background like stability of IT-Environment, computational effort and definition of constraints will flow into current car projects and have to be adapted to their more complex requirements.

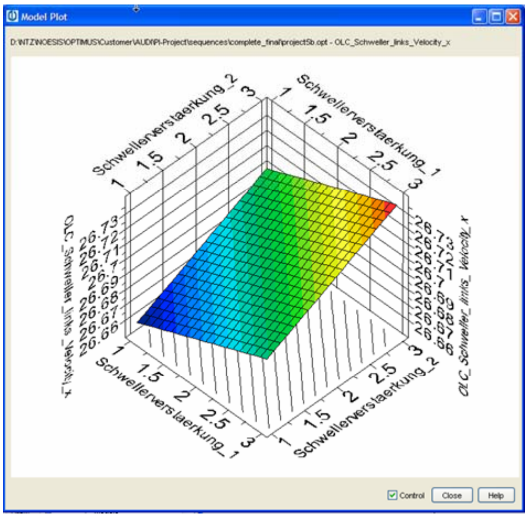

Fig. 5: Response Surface Model

More load cases and much more constraints have to be considered to execute a reduction in the weight of a whole car using such an optimization in a real design phase. Therefore, more resources and more effective algorithms are required. Under the existing limitation of resources further projects were executed on local problems of components.

3.2. Optimization of the Restraint System for Occupant Safety

In modern restraint systems many components and their interaction are responsible for achieving an optimum design in occupant safety. The complete system must provide a nearly optimal behaviour for every occupant in all possible accident conditions: May the person be small or tall, belted or unbelted and exposed to different velocities, angles and crash configurations.

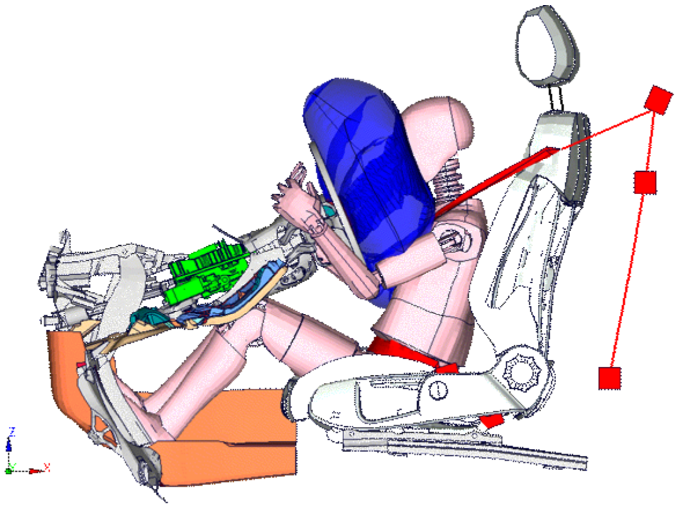

Fig. 6: Restraint system for Occupant safety

The typical question for an optimization tool is to find an optimal parameter set for all situations. Doing this “by hand” is an almost impossible task. Thus, the use of multidisciplinary optimization becomes a must. Here, 4 load cases were considered and 10 parameters were allowed to modify. Due to the experience obtained in the first project the actual design was used as start design. In this design the parameters were determined with plenty of stochastic simulations however the design still violated a few of the 16 constraints. Therefore, the aim was to find a design which fits all constraints like knee forces, head injury criterion (HIC) or chest deflection and to minimize the chest acceleration of the different occupants. Additionally, some arrangements were added like redundant licence server to be insensitive to a breakdown of a server or other instabilities.

For this optimization 4 generations with 20 designs each were necessary for achieving a good design. Altogether 320 simulations were computed within 7 days using a mean number of approx. 160 CPUs.

Afterwards, a surrogate model was created to search for further improvements. The design after the RSM was slightly improved. After verifying these parameters they were directly adopted as a solution in the car project.

3.3. Optimization for Pedestrian Protection

Another project concentrating on the demands of pedestrian protection was carried out. In this case just one load case was considered but a new additional challenge was implemented. Two outputs where used for defining the objective. Material properties where used as design parameters and additionally geometry modifications where allowed. TOSCA was used for the morphing of the solid mesh of one part. As constraints the maximum value of the vectors for the knee bending angle and for the acceleration of the lower leg were considered.

The pretentious task in this project was not just to find one specific design but also to see the effect of combined changes in parameters, material properties and mesh geometry. It was more important that the results gave a feeling of the correlations between these parameters (inputs and outputs).

The generated surrogate model can be used in project meetings for a rough estimation of the effect of design changes. Any changes of the input parameters can be adjusted directly and the consequences can be communicated “online”. The response time decreases as some decisions can be taken without further simulation loops.

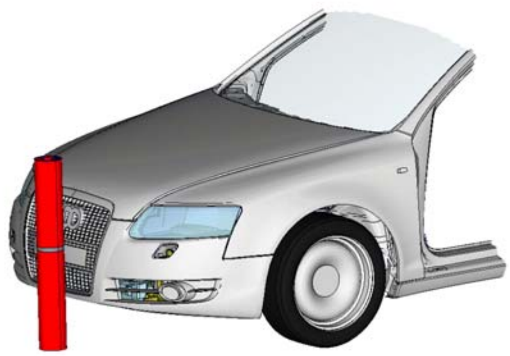

Fig. 7: Pedestrian safety

4. Summary and Outlook

The employment of an optimization tool is a good means to solve complex issues with a minimum of interactive work. Provided that the costs for computation time decreases, more problems will be able to be solved by a multidisciplinary optimization approach. Today, due to the limited number of CPU’s a “daily use” of this tool for all problems is not yet possible. Improvements of optimization algorithms and their clever combination will also help to extend the use of MDO.

Beside the problem of the limited number of CPU’s, the graphical interface needs to be very intuitive and the choice of optimization algorithms and the associated parameters must not be too complicated. Further improvements in the graphical user interface and the interfaces to the analysis software will help to reduce the knowledge and the effort necessary for the setup of an MDO process. Furthermore this will help to enable every engineer, and not just the MDO specialist, to use this tool. Especially robustness and stability of the process are very important to guarantee that the process will run for several days even though plenty of problems in the IT-system might occur.

The users of MDO will have to invest more time to improve the formulation of objectives, constraints and targets. Since the quality of the answer is dependent of the quality of the questions, the engineer has to ensure that the automatic evaluation of simulated designs works as well as if done “by hand”.

The employment of MDO at AUDI AG is a helpful enhancement for the work in current and future car projects. The work with MDO will be continued and extended.

5. References

[1] N. Tzannetakis, P. van Vooren, B. Lauber: The Use of OPTIMUS for Advanced Multi Disciplinary Structural Optimization in Automotive Applications; NAFEMS Seminar Wiesbaden, 27.-28. April 2005

[2] Noesis Solutions NV: OPTIMUS User's Manual. Leuven, Belgium (2004)