

CUSTOMER STORY // TURBO MACHINERY

# FE-MODEL GENERATION OF TURBOCHARGER BLADES REGARDING GEOMETRICAL TOLERANCES

With the help of optiSlang and SoS, realistic FE-models of turbine wheels as a part of small gasoline turbochargers were generated in order to analyze and to optimize their tolerance behavior.

## Introduction

The goal was to develop a fully automatized procedure to generate 3D-CAD geometries of turbocharger turbine wheels including different kinds of real production imperfections. The procedure incorporated the evaluation of particular deviations and differences from nominal blade geometry, hub body geometry and backface geometry. Since the turbine design was integrated, the blades and hub have been considered as a single part.

The wheel was manufactured by investment casting, so different sources of deviations were considered. Tool tolerances, casting process parameters, shrinking of wax and metal during solidification and cooling as well as finishing process steps had influence on the final geometry. Each geometrical feature, like massive hub body, thin blade body, machined or un-machined surfaces had different deviations. In the numerical system, the process of determination of deviations could be reproduced for many different virtual geometry designs and the space of the designs' deviations could be statistically evaluated. Based on these statistical evaluations, it could be stated with quantified probability in which interval ranges the geometrical deviations occurred.

The original numerical simulation process for turbine wheels design was split up into geometry generation and FEM analysis. Geometry generation needed to be parametrized to set up an automatized repeatable design generator. Utilizing a progressive technology of statistical metamodeling implicitly included in optiSlang, a statistical Metamodel of Optimal Prognosis (MOP) describing relations between input parameters (geometry modification) and output parameters (geometry deviations) could be established. Using such a statistical metamodel as a solver instead of a geometry generation process, the whole procedure could be increased rapidly.

The incorporation of virtual simulations of geometrical deviation into the process of turbocharger development had a certain positive impact on a better understanding of the deviation causes and deviation statistical properties. This knowledge led to a better performing turbocharger design and eliminated unnecessary tight tolerances. On the other hand, the robustness of several design features could be evaluated and improved.

## Contact-element based algorithm for tolerances evaluation

Algorithms for the evaluation of deviations represented the core of the whole process. Four different algorithms were designed to measure four different types of deviations. Assuming production deviations, it is necessary to measure the distances between external surfaces (see Fig. 1), thickness differences, curves and points distances. ANSYS classic environment was chosen for the implementation of algorithms ensuring robustness and a wide variability in customization. Contact and target finite elements (designed and derived in ANSYS for performing nonlinear structural analysis) were used to determine distances between the defined surfaces (gaps respectively penetrations in terminology of ANSYS). Based on this feature, differences between the nominal and design geometry were calculated and further processed. The results were available for all nodes of the FEM mesh, but it was advisable to pick a certain number of relevant nodes for the evaluation of production scatter. Interesting post-processing nodes might be located either on the edges that could be measured with tactile instruments or on topological points that could also be checked by optical measurement systems. To be able to understand the system behaviour, a reasonable amount of nodes needed to be selected intelligently. Full surface results were nevertheless an interesting source of information when selecting designs for further analysis. Through the use of numerical contact algorithms and distance calculation, they resembled the typical post-processing results of optical 3D scans.

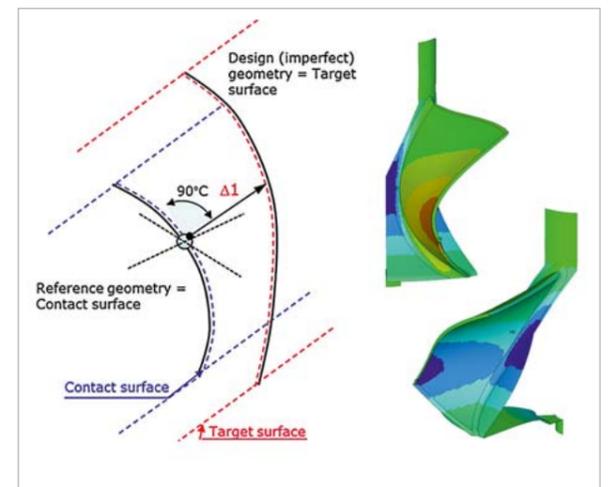


Fig. 1: Example of external surface deviations on the blade and hub body

tions from the moment of geometry creation in BladeGen and DesignModeler until the deviation of automatic value extraction. It was the only way to post-process the results from hundreds of different designs. Some design features like blade thickness were exclusively defined in BladeGen, while others like fillet radii were exclusively defined in DesignModeler. Both systems have their own interfaces and file formats. Also, in case of impossible geometries, different exit conditions had to be recognized. These inherent properties of the task made it necessary to have a generic control system for the numerical process chain. The key control system determining the time flow of the process is optiSlang4. It enabled the user to compose a sophisticated structure of particular actors representing the various actions that were supposed to happen during the flow run (see Fig. 2).

## Process integration

As the core of the process was the deviation measuring techniques prepared in the classic environment of ANSYS, it was a crucial task to set up the process of gaining the devia-

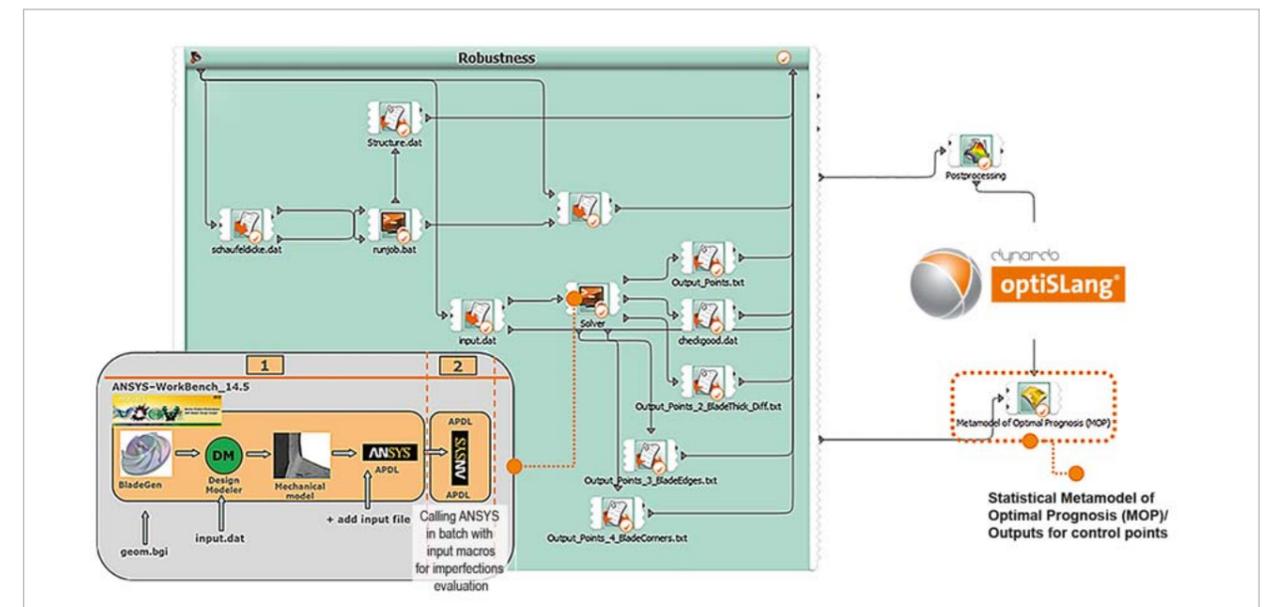


Fig. 2: Process flow in optiSlang

The process started with the creation of the correlated input set of parameters. Parameters were spatially correlated using the random fields' technique (see Fig. 3). Correlation dependencies designed by random fields secured that the geometrical deviations result in "reasonably" imperfed blade designs (see Fig. 3). Designed blade surfaces with higher density of surface waves were not in compliance with the produced turbochargers. Other parameters like blade length were generated randomly.

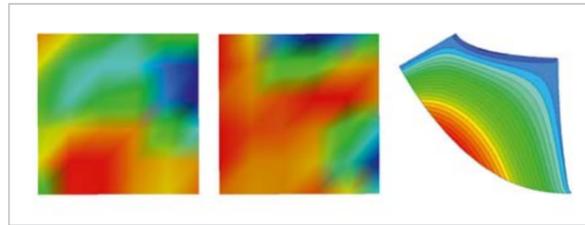


Fig. 3: Example of random fields' realizations (left), correlated thickness distributions (right)

After the preparations of input parameters, the main part of the process was started (see Fig.2 – main solver part). A new blade design was produced by BladeGen based on the correlated input parameters and other parameters passed on into the BladeGen tool. The blade was then finished in DesignModeler, connected to the hub body with a fillet radius, fitted with a backface and nose as well as prepared for exporting to ANSYS solver. Deviations were calculated using ANSYS and sent to optiSLang4 as responses. optiSLang4 evaluated the statistical quantities and created a MOP for the chosen responses.

**Strategy of producing non-nominal geometries**

The process described in chapter 3 could be performed as a sensitivity (robustness) analysis. Results of such a procedure were the statistical quantities representing the dependencies between the input and output parameters. Since not all parameters of the geometry generation related directly to a length or position, it was useful to have a tool that quantified scatter of the actual measured feature versus the input parameter. This was necessary information when tuning the deviations to typical manufacturing

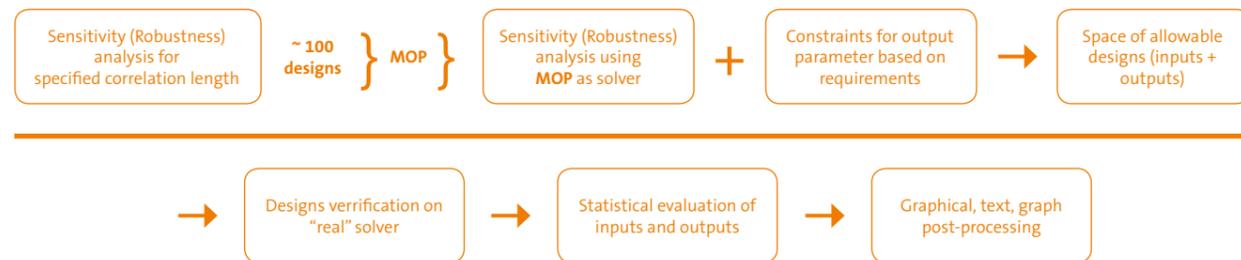


Fig. 5: Strategy for the evaluation of deviations

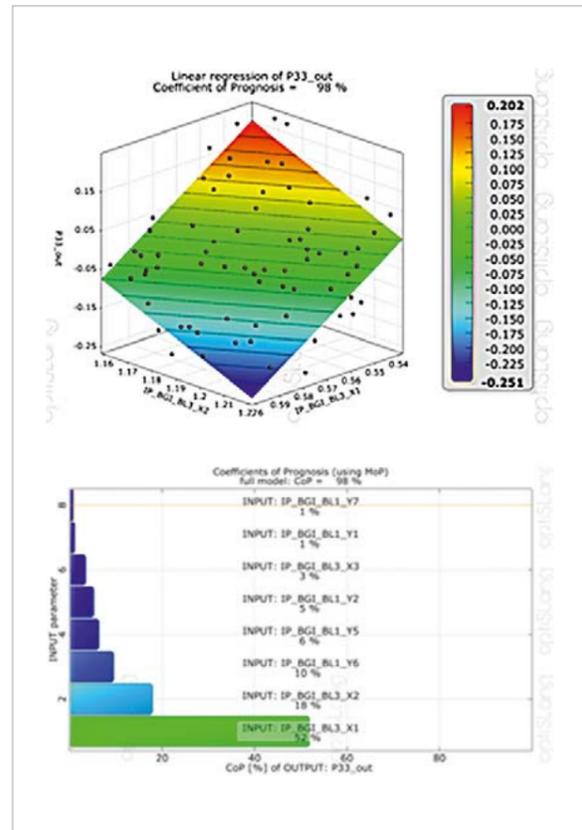


Fig. 4: MOP for the chosen response (top), Coefficient of Prognosis (bottom) The process flow integration was based on Windows and Python scripts.

values, especially when splines were used in geometry generation. Over these statistical quantities, the MOP could be created for more important purposes:

1. Quantification of the explainability of the output parameters.
2. Determination of the dependencies between input and output parameters.
3. Statistical verification of the deterministic procedure.
4. MOP could be used as a substitutive solver

Utilizing MOP as a solver, it was possible to calculate a sufficient amount of designs in a reasonable time. Designs calculated this way were cross-checked by the parallel per-

forming of the full process. After collecting all the responses, different response filters were applied to create a space of allowable designs.

**Sensitivity analysis**

The sensitivity analysis provided basic statistical properties of the inspected problem. As a first step, a set of input and output parameters had to be defined. In between, the functional dependencies were expected. Using LHS, it was possible to cover the desired design space (within the input parameters' ranges) with a reduced number of samples (50-200, see Fig. 6).

An evaluation of structural mechanics of the non-nominal designs expected extreme cases to be the most interesting ones. Therefore, a non-centrally emphasized sampling was helpful. This was even more relevant when high nonlinearities were involved. The effect of one parameter might have been much higher in a border area of the design space than in the center or on the opposite side. Information about this could only be available when the sampling combined boundary values of several parameters at the same time. The higher the number of evaluated samples, the better quality of statistical properties was to be expected. The dependence of the number of input parameters was low, but with a number of around 50 input parameters, it was advisable to do at least 100 successful designs with LHS. To be able to achieve this, even under the presence of instabilities, a larger number was requested in optiSLang accordingly. The run could be aborted when the number of successful designs was reached. Performing sensitivity analysis, the following valuable information was provided:

1. Stability of designed process workflow (eventual manifestation of conflicts)
2. Relations between input and output parameters were determined
3. Utilizing the MOP on the design space, it was possible to determine the importance of the input parameters on each of the output parameters. Additionally, the participation of the input parameters was quantified. Dependencies determined between inputs and outputs could be highly non-linear as well.
4. Obtaining high values of Coefficients of Prognosis (CoP) for the responses, it is proven that defined responses can be well explained by the defined input parameters. In an opposite case the reasons for low values of CoPs should be considered. This way, the whole process was subjected to statistical verification.
5. MOP represents the mathematical dependencies between the inputs and outputs. Knowing these dependencies, it was possible to use such a statistical metamodel as a substitutive (significantly faster) solver. Results obtained from such a solver contain a certain error expressed by the CoP.

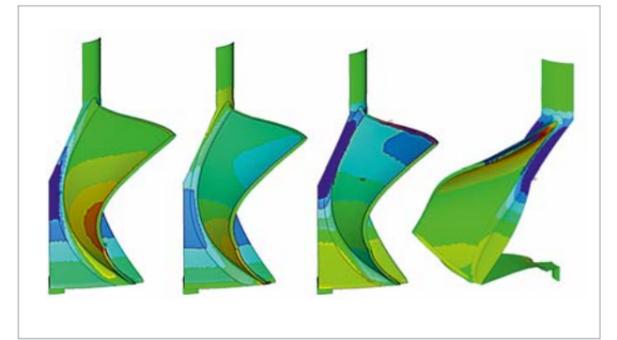


Fig. 6: Variability of the parameterization

Sensitivity analyses were successfully carried out either by using the whole designed procedure or using the MOP as a substitutive solver.

**Metamodel of Optimal Prognosis (MOP) as a generator of non-nominal geometries**

The MOP is a statistical meta-model containing special features suitable for usage in a wide spectrum of probabilistic problems. As any statistical model, it is able to predict the values of responses with a certain quality of approximation. The prediction quality of an approximation model may be improved if unimportant variables are removed from the model. This idea is adopted in the MOP which is based on the search for the optimal input variable set and the most appropriate approximation model (polynomial or MLS with linear or quadratic basis). Due to the model independence and objectivity of the CoP measure, it is well suited to compare the different models in the different subspaces. As it is possible to reach a high precision of the MOP (quantified by COP), it is very convenient then to use it as a substitutive solver representing dependencies between input and output qualities. In the case of the presented calculation process of a turbine wheel's deviations, it took about 25-30 minutes to complete one design containing unique geometry variation. The main fraction of this time was used for distance calculation between thousands of nodes. But the geometry generation in DesignModeler was also costly due to the interface with BladeGen on the one hand and 3D fillet generation on the other hand. After solving a sufficient amount of various wheel designs (in this case ca. 120) and building up the metamodel over the design space, it was stated that over 90% of the output parameters had a COP higher than or equal to 85% (see Fig. 4). Based on this knowledge, it was feasible to use the MOP as a substitutive solver with the expectation of obtaining a reasonable quality of results. Utilizing the MOP as a solver in the process workflow caused a dramatic acceleration of design generating performance. Compared to the full process workflow, the speed when using the MOP was more than 1000 times faster. Due to such an acceleration, it was possible to carry out sensitivity (robustness) analyses containing 2000 designs and more in less than one hour. This performance sig-

nificantly gained a higher amount of designs than it would be possible with only a full workflow. It brought to light valuable statistical information about the relations between the geometry variations and appropriate deviations.

### Filters

One of the consequences resulting from MOP utilization was the higher amount of produced output data. To get an overview of design scatter, many ways of data post-processing exist. Histograms of frequencies of occurrence (Fig. 7) can be displayed for each of the output parameters. Each histogram can be approximated by the best-fitting type of statistical distribution. Once the statistical distribution was attributed, the probability of response occurrence in a specified continuous interval could be easily determined. Sorting the output of designs according to the chosen criteria was a way to aggregate the result information from the whole design space. In optiSlang4, it is convenient to use the constraint conditions feature in order to sort or to filter the designs according to ranges of response. Intending to implement a filter which will sort out all the designs having at least one of the responses (from a selected set of appropriate responses) out of a given interval (symmetric, defined by bound =  $\beta$ ), it is necessary to set up the following conditional constraining equations for all involved responses:

$$ABS(Value_{Desing}) \leq Value_{required} \rightarrow ABS(P18\_OUT\_Th\_Diff) \leq \beta$$

$$\rightarrow ABS("all\ other\ related\ responses") \leq \beta$$

By the application of the formerly described filter on the design space, only the designs fulfilling the conditions for all the responses remained. The others were considered to be invalid. The primary deficiency of this basic filtering technique was the fact that a design could only be valid (status=1) or invalid (status=0). Furthermore, neither the amount of responses that violated the allowable bounds for each design nor their extent of violation was known. In order to obtain

this information for the estimation whether the violation was only local or occurred at a larger area, it was necessary to create a new actor in optiSlang4 that contained a Python function summing up the violations for each design. The advantage was a deeper insight into the probability of the occurrence of limit violations. An example of another useful filter is the “two belts filter”. Also, a certain tolerance on the allowed deviations of 20% was introduced. The purpose of this filter was to sort out all the designs with responses outside the two defined intervals (see Fig. 8).

$$\rightarrow ABS(\text{All related responses}) \geq 0.8 \times \beta \wedge (\text{All related responses}) \leq 1.2 \times \beta \wedge (\text{All related responses}) \geq -1.2 \times \beta$$

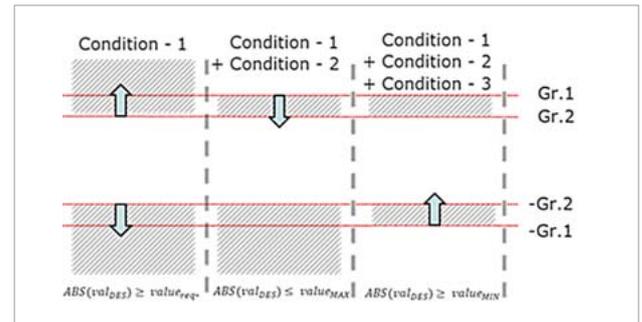


Fig. 8: Example of “two belts filter”

The filters based on constraining equations could be implemented either in the GUI, using predefined interface, or in Python scripts that could be prepared in advance and later on be inserted in optiSlang4. The opportunity of using Python scripts at any moment of creation of an optiSlang4 workflow enabled a preparation of a higher amount of responses, conditions, parameters etc.

**Author //** Dr. B. Lehmayr (Continental Automotive GmbH) / M. Mrozek, Dr.-Ing. R. Schlegel (Dynardo GmbH)  
**Source //** [www.dynardo.de/en/library](http://www.dynardo.de/en/library)

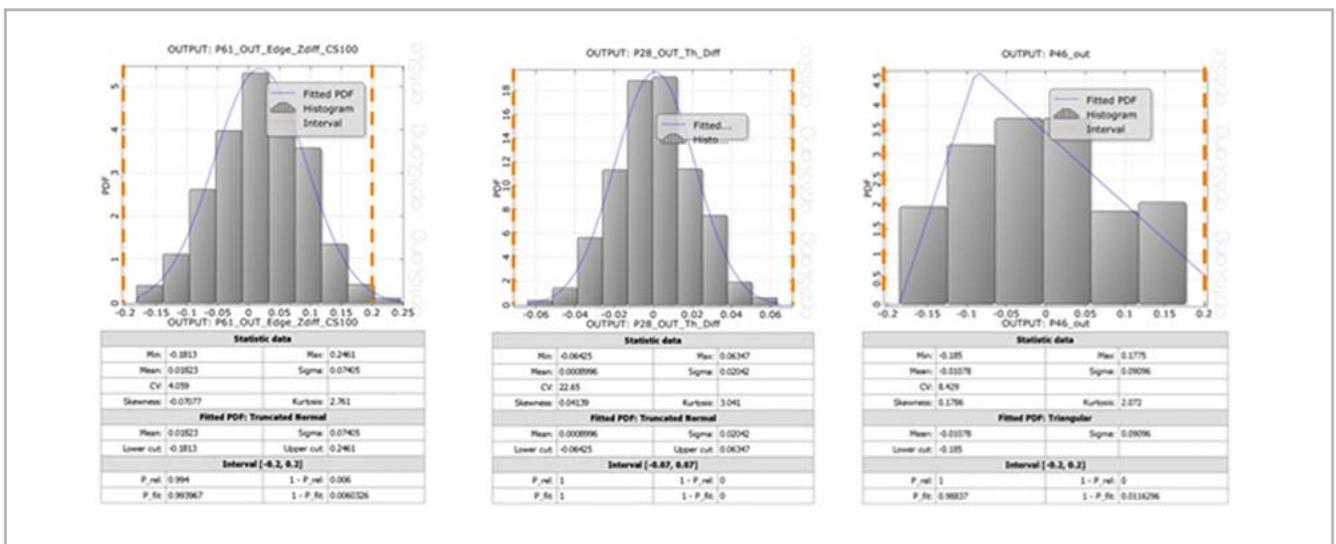


Fig. 7: Statistical distribution of attribution