Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-45618

HIGH-DIMENSIONAL CONSTRAINED MULTIOBJECTIVE OPTIMIZATION OF A FAN STAGE

Marcel Aulich and Ulrich Siller
German Aerospace Center
Institute of Propulsion Technology
Fan and Compressor Department
51147 Cologne, Germany
marcel.aulich@dlr.de ulrich.siller@dlr.de

ABSTRACT

A high-dimensional design space, different objectives, many constraints and a time-consuming process chain result in a complex task for any optimization tool. This paper shows methods and strategies used at DLR, Institute of Propulsion Technology, to handle this kind of problem.

The present optimization task is a rotor-stator configuration with more than two hundred free design variables, two objective functions (efficiency, stall margin) and mechanical and aerodynamic constraints (mass flow, eigenfrequencies, etc.). The process chain consists of geometry and mesh generation, FEM- and 3D-CFD calculations for different operating points.

After defining the setup and explaining the initial already 3-D-preoptimized configuration, the CFD/FEM optimization tool is described. This tool calculates the complete CFD/FEM process chain and creates new designs (also called members) by using an evolutionary algorithms.

Parallel to the CFD/FEM optimization a program based on surrogate models is running. By using surrogate models a fast evaluation of new members is enabled. So a database of new members can be created quickly. Based on this database a set of new members is built. This is send to the CFD/FEM optimization tool, where the complete CFD/FEM process chain is applied. After the CFD/FEM evaluation process, these member are used to train the surrogate models again. This procedure repeats until the optimization goals are reached.

In the next part of this paper the implemented surrogate models are discussed. Both neural networks and Kriging models have advantages and disadvantages compared to each other. It is important to understand them to choose the right model at the right time of optimization.

The main focus of this paper is on the selection criterion for new members. This criterion has two targets: push the performance of the fan stage and enhance the surrogate models. At first sight these targets seem to be contrary, but the surrogate models do not predict a single mean value for an objective. They offer a density distribution of the potential objective values. That allows calculation of the Paretofront enhancement (ParetoEnSet) for a set of new members. ParetoEnSet is the expected area gain of a set of members to the current Paretofront. This criterion based on the already known expected improvement. It is shown, that ParetoEnSet can rise, when the uncertainty of an prediction increases. The uncertainty is estimated by a surrogate model. So new members tend to explore the design space, where the predicted uncertainty is huge. These members are favorable for improving the surrogate models. In addition, it is easy to couple constraints with ParetoEnSet.

In the last section the results of the optimization are illustrated. Compared to baseline design the optimized stage accomplishes a notable improvement in efficiency by obtaining the stall margin and fulfilling multi aerodynamical and mechanical constraints.

Nomenclature

η_{is} = isentropic efficiency

AL100 = operating line at 100 percent rotational speed
AL79 = operating line at 79 percent rotational speed
CFD = computational fluid dynamics
DLR = Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
ExpImpr = expected improvement
ExpVolGain = expected gain of volume
FEM = finite element method
MLH = maximum likelihood
NS100 = near stall at 100 percent rotational speed
NS79 = near stall at 79 percent rotational speed
ParetoEn = Paretofront enhancement
ParetoEnSet = Paretofront enhancement of a set of members

Introduction

At the Institute of Propulsion Technology a couple of complex optimizations were successfully performed in the last years (e. g. see [12],[11],[6] and [5]).

This paper illustrates recent optimization strategies, which have proven their relevance in practical turbomachinery design. The performance of the optimization tools is demonstrated on a sophisticated fan stage design.

Starting Situation

The optimization object is a transonic low bypass fan stage consisting of a single rotor and a tandem-stator designed to provide a very high pressure ratio (see [1] and [6]). Both rotor and tandem-stator are aerodynamically highly loaded. Due to low stage reaction ($\rho_{hub} = 0.25$) in the hub part the stator decreases the high inflow Mach number of 1.2 to 0.55 towards the exit. There were two important 3D-optimizations of this stage in the past: In 2008 a full stage optimization was conducted with overall 230 free design parameters at the annular duct, rotor and tandem-stator section profiles and stacking. Four aerodynamic operating points at two rotational speedlines allowed consideration of working line performance and surge margin control. The resulting stage design showed significant improvements of aerodynamic performance, but a second optimization was necessary due to problems with blade statics. This multi-disciplinary approach also included rotor statics and the Campbell diagram [6].

Subject of the current optimization is the remaining potential seen for the stator with a refined set of degrees of freedom and consideration of the stator dynamics, namely the Campbell diagrams.

Free Variables

The optimization was conducted with a total number of 210 free geometric parameters. This set of free design variables was chosen to provide advanced geometrical modifications targeting on further improvement of the already pre-optimized stator configuration. Figure 1 shows in a meridional view the annular duct contour and the blades of the baseline member. In addition, an overview of the chosen parametrization is given. The duct hub and casing contours are free in the stator region, the relatively fine distributed spline control points are free to shift in radial direction. The axial positioning and axial blade length of the two stator rows are parameterized by a set of control points in the meridional plane. All of these points (orange circles) except the Stator I leading edge point at hub and the Stator II trailing edge are free for optimization. This enables a redistribution of axial chord length between the stators as well as axial overlapping configurations. The stators are generated using five profile sections each. Pitchwise shifts of these profiles provide optimization of the relative positioning of the two stators and 3D-stacking features like a “bow”. As seen in the box in figure 1 the profiles are parameterized using the classical profile angles and spline control points for the suction side. The pressure side is defined by a thickness distribution - as offset from the suction side - for a given maximum profile thickness and its position as well as area criteria (“fillFactor”). The full set of these parameters is used as free variables in the optimization. The blade count (same for both stators) is another important free design variable.

Simulation Setup

The process chain, which is to be executed in the scope of optimization consists of the following steps/tools:

* Geometry Generation: The annular duct and the blades are generated using in-house tools.
* Mesh Generation:
  - FEM: A structured, solid mesh of the stator blades is generated for the finite element analysis. This mesh has about 15000 cells for each stator.
  - CFD, see figure 2: The full stage including the invariant rotor is meshed to capture the overall aerodynamic performance and maintain the proper matching. The in-house mesh generation tool “G3dmesh” is used for the structured multi-block rotor mesh. For the stators, an in-house tool “GTA” calculates for a given pitch the periodic boundary surface. A hybrid structured/unstructured mesh accounts for the complex geometrical situation in the stator part. The unstructured mesh is generated using the commercial mesh generation software CENTAUR. This allows robust and high quality meshes for highly different stator configurations (e.g. axial overlapping) but results in much larger setups with respect the number of cells. A fully structured mesh of the stage

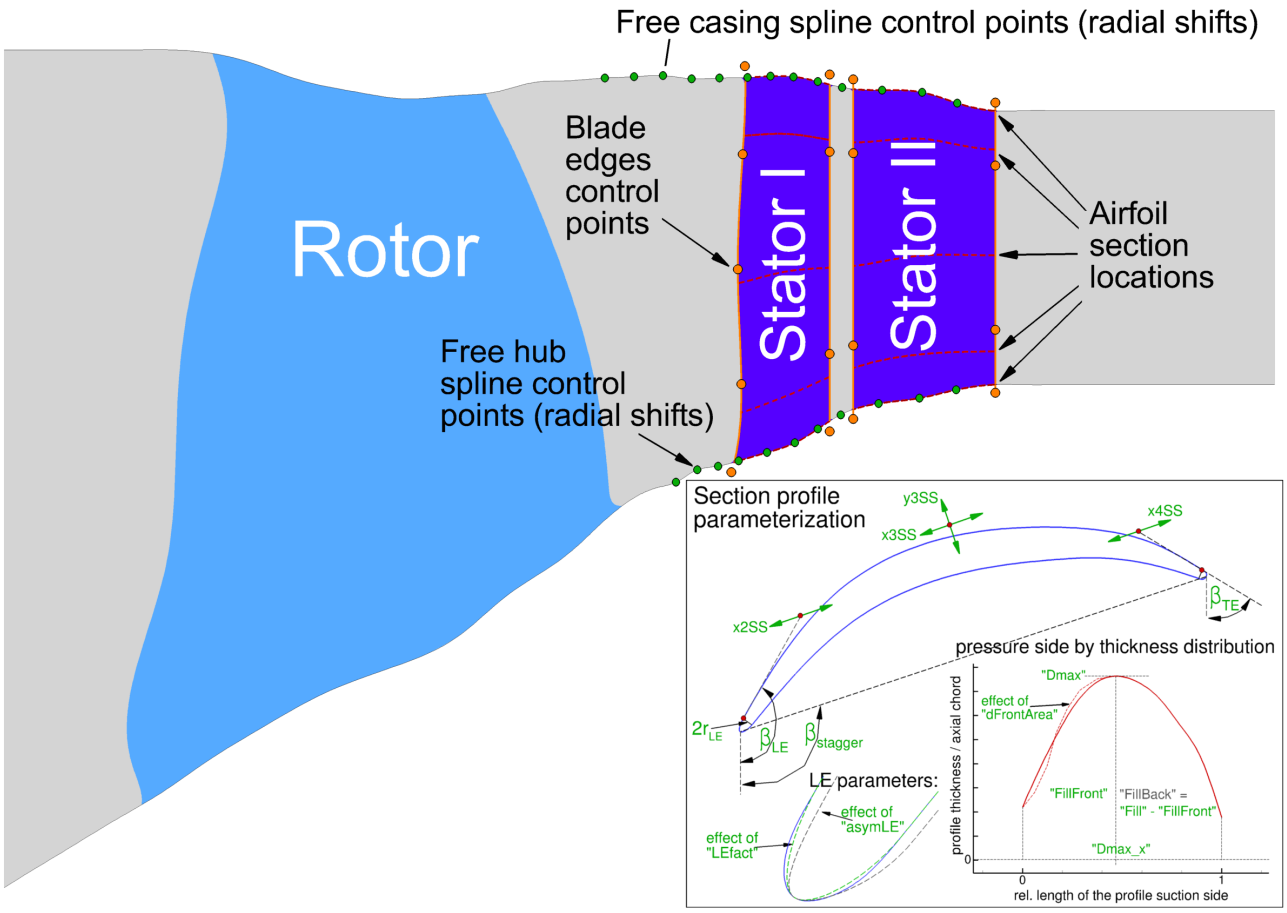

Figure 1. Parametrization and free design variables

has about 1.5 million cells, the optimization hybrid mesh has 4.5 million cells.

* Simulation:
  - FEM: The stator eigenfrequencies are evaluated using the open-source finite element tool CalculiX (see [2]). Constant static pressure surface loads of the initial design has been considered due to the focus on the blade eigenfrequencies. The complex mechanical boundary conditions of the stators (fabrication with selective laser melting (SLM) as clusters) are not reflected in the numerical setup with firmly clamped stator endwalls. These effects are considered by correction offset values to the eigenfrequencies, which were determined prior to optimization.
  - CFD: Overall four aerodynamic operating points are chosen: Compressor working line and near stall for two rotational speeds. A constant mass flow rate is set for the two near stall operating points using a PID-controller inside the flow solver adjusting the outflow static pressure. The steady 3D-RANS computations are conducted using DLR's turbomachinery method TRACE [3][4].

Objectives and Constraints

As optimization objectives o_1 and o_2 the sum of the working line isentropic stage efficiencies and a stall margin criterion is used :

$$o_1 = \frac{1}{2}\left(\eta_{is,AL100} + \eta_{is,AL79}\right)$$

$$o_2 = \frac{1}{2}\left(\frac{\Pi_{tot,NS100}}{\Pi_{tot,WL(\dot{m}_{NS100})}} + \frac{\Pi_{tot,NS79}}{\Pi_{tot,WL(\dot{m}_{NS79})}}\right)$$

$\Pi_{tot,WL(\dot{m}_{NS100})}$ and $\Pi_{tot,WL(\dot{m}_{NS79})}$ are the working line total pressure ratios at the same massflows $\dot{m}_{NS100}$ and $\dot{m}_{NS79}$ like the near stall points NS100 and NS79.

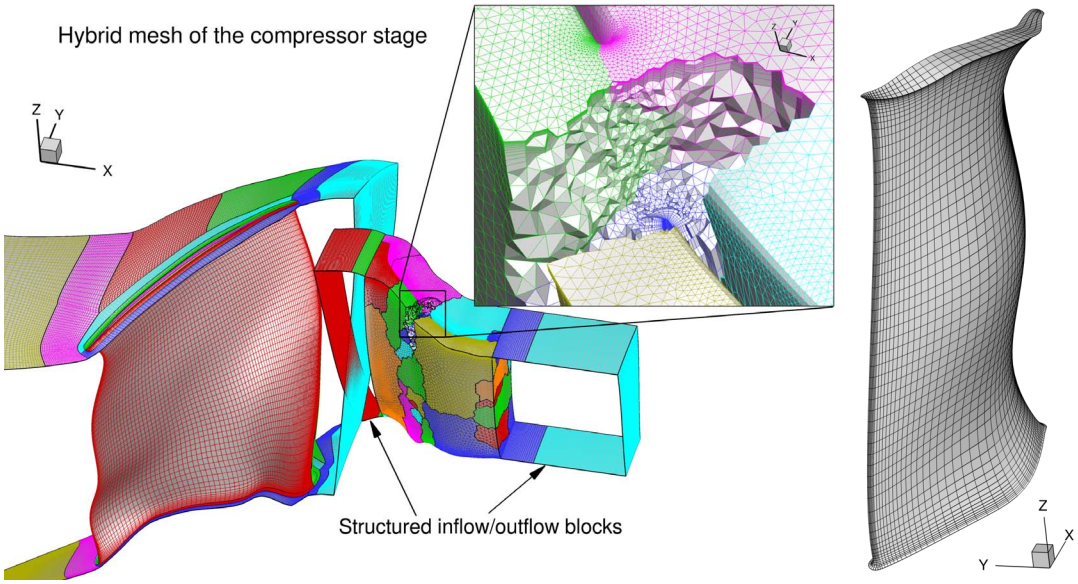

Figure 2. Hybrid structured/unstructured mesh (CFD) and solid mesh of the second stator (FEM)

As aerodynamic constraints the massflow rates in both working line operating points and the stage exit swirl angle has been considered. The overall exit swirl deviation is calculated by radially integrating the mass weighted absolute deviation of the stage outflow swirl angle from the targeted value.

To avoid forced response issues, the lower eigenfrequencies of both stators are used as constraints to avoid crossings with the first rotor harmonics in the Campbell diagrams.

Optimization

The optimization is performed by the optimization framework AutoOpti (see figure 3). AutoOpti has been developed at the Institute of Propulsion Technology over the past eight years with a focus on turbomachinery applications.

AutoOpti consists of two programs running in parallel.

CFD/FEM Optimization

The CFD/FEM optimization is displayed on the right hand side of figure 3. The program is based on an asynchronous evolutionary algorithm. Hereby created members, sets of the free parameters, are communicated to the slave processes. Every slave performs the CFD/FEM process chain for one member. The results, fitness and other objectives of interest, are returned to the root process.

Acceleration Process

The acceleration process, displayed on the left hand side of figure 3, is based on surrogate models (also called metamodels).

The surrogate models are used to approximate the objectives of the optimization. To cut down on process time, the fitness functions as well as the constraints are reproduced by surrogate models. This enables a fast evaluation of new members.

Kriging and Bayesian neural networks are applied as surrogate models. Both provide an uncertainty estimation of their predicted values. This is an important fact for the selection of new members.

The acceleration process is an iterative program repeating the following steps:

At the beginning of every loop the current CFD/FEM optimization database is taken. This database is used inside the surrogate model for training. Kriging, neural networks or both can be trained.

Next a genetic algorithm is applied to create a database of new members. This database is created in a short time, because the time-consuming parts of the process chain are replaced by the trained surrogate models.

Finally a set of new members is selected and communicated by an interface to the CFD/FEM optimization. There the new members pass through the complete process chain including the CFD and FEM simulations for final validation .

For choosing a surrogate model, it is important to know the advantages and disadvantages of Kriging and neural networks.

Kriging

Ordinary Kriging is one surrogate model inside AutoOpti. The estimated fitness $\hat{y}$ at location $\overrightarrow{x}$ is given by:

$$\hat{y} = \hat{\beta} + \vec{r}^{T}(\overrightarrow{x}) \cdot \overleftrightarrow{R}^{-1}(\overrightarrow{\theta}, \overrightarrow{p}) \cdot (\overrightarrow{y_s} - \hat{\beta} \cdot \overrightarrow{1})$$

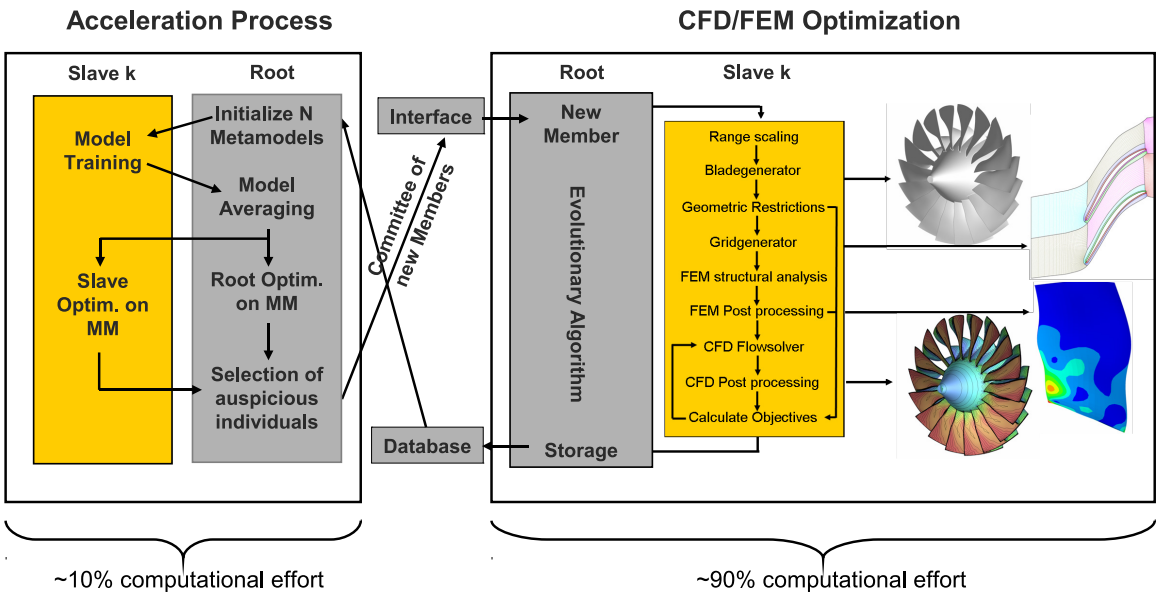

Figure 3. optimization framework *AutoOpti* - the basic flowchart

$\overleftrightarrow{R}$ is the correlation matrix of the database members, $\vec{r}^T(\vec{x})$ is the correlation vector of $\vec{x}$ to the database and $\vec{y_s}$ are the fitness values of the database members. In n-dimensional design space the correlation $corr(\vec{x_1},\vec{x_2})$ between two datasets $\vec{x_1}$ and $\vec{x_2}$ is:

$$corr(\vec{x_1},\vec{x_2}) = exp^{-\sum_{i=1}^{n}\theta_i|x_{1,i}-x_{2,i}|^p}$$

$\theta_i \geq 0$ is a scaling factor of parameter i and $p \in [1,2]$ controls the curve shape. A component k of $\vec{r}(\vec{x})$ is the correlation between $\vec{x}$ and the variables $\vec{x_k}$ of the kth member:

$$r_k(x) = corr(\vec{x},\vec{x_k})$$

An element $d_{i,j}$ of correlation matrix $\overleftrightarrow{R}$ is the correlation between members i and j:

$$d_{i,j} = corr(\vec{x_i},\vec{x_j})$$

An advantage of the Kriging model is its ability to reduce the number of effective parameters, if the database is small. In this case some θ_i-values are set to zero. So the corresponding parameters have no influence in the fitness calculation of $\hat{y}$. This property allows to adopt a Kriging model very early in the optimization process. Even, if the number of training members is less than the number of free variables, the adoption of a Kriging model can be successful, because usually there are some dominant variables in high dimensional design space. Hence new members can be created by optimizing only these dominant variables. This attribute is often denied, because if the number of training members is less than the number of free variables a linear interpolation can be accomplished.

Contrary to other Kriging models, ordinary Kriging evaluates a member with the mean value $\hat{\beta}$ of an objective, if the member requires too much extrapolation compared to the training database ($\vec{r}(\vec{x}) = 0$). This is important, because in high-dimensional space extrapolation works only in a very small range.

Compared to a neural network it is another benefit of a Kriging model, that it is directly coupled with the database (it interpolates the database even without trained model parameter). So the approximation is rarely unusable. For the same reasons the complexity of a Kriging model is limited by the number of trainings member. Contrary neural networks can become too complex, if there are too many weights used.

The disadvantage of a Kriging model is, that the time for performing a model parameter training rises with $O(n^3)$ (n is the number of training members) when it is MLH optimized. So it is not possible to use Kriging with a big database (many thousands of members). Another problem are similar members in the database. In this case the MLH optimization can become ill-conditioned.

In our case a model parameter (θ_i) for each variable is used to calculate the correlation between two training members. So the correlation function is axis symmetric. This implicates, that the optimized Kriging model is usually changing, if the training members are rotated in the design space, because the required

correlation function can not be reproduced, see figure 4. By

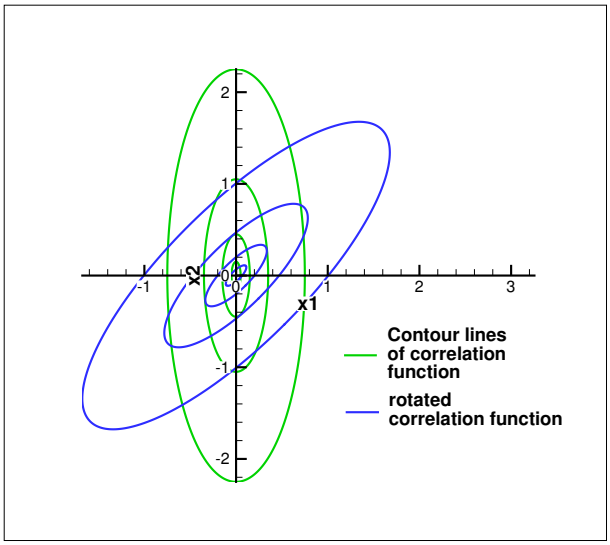

Figure 4. The correlation function $corr(\vec{x},0)$ is axis symmetric. Correlation functions like the blue one can not be replicated.

rotating the training members, distances between members are not changing. So the optimal Kriging approximation should be preserved, only rotated in the same way.

Rotating the design space is only a theoretical approach, but it shows that Kriging does not depend only on the distances between the training members. Except a theta matrix is used, but then the number of model parameters would increase quadratically.

Neural Network

The implemented neural networks model is a Bayesian feedforward network with automatic relevance determination (see [9] and [10]). The training algorithm based on the "weight-decay" approach:

$$\begin{aligned} min! : F &= F_D + \lambda F_W \\ &= \frac{1}{2}\sum_{i=1}^{D}(f(x_i) - y(x_i,w))^2 + \lambda(\frac{1}{2}\sum_{j=1}^{W} w_j^2) \end{aligned}$$

The concept of Bayesian neural network is to estimate the regularization parameter λ of the "weight-decay" method by a probabilistic approach. So the training searches for the best compromise between model complexity F_w (quadratic sum of all weights) and data approximation F_D (mean squared error between real fitness values $f(x)$ and network output $y(x,w)$). This is comparable to bias-variance tradeoff.

The time of a neural network training depends primarily on the network structure (number of layers, knots and weights). Increasing the number of training members yields a linear increase of training time. This enables the consideration of a huge training database.

For a Kriging training, in contrast, the number of members has to be reduced. In this case the problem is to find members with redundant information (if these members exist), which can be removed from the database. This selection is often performed by simple algorithms, which could choose an unsuitable training database.

But a neural network, whose complexity is comparable to a Kriging model with a reduced database (that means the number of weights is roughly the number of members in the reduced database), finds the redundant information in the database by its training algorithm. The training shifts the weights of the neural network to the area, where the complexity of the fitness function increases. So no member selection is needed before training.

Also similar training members, even the same member multiple times in the training database, have no direct influence on the condition of the training algorithm.

The big task in training neural networks is to find an adequate structure and initialization (initial setup). The initial function of a neural network is more or less randomly before training. By changing the structure and initialization primarily the complexity is influenced. Hence at the beginning of the training the initial situations significantly differ. This is different to Kriging models, where the initial structure is fitted to the database. Compared to Kriging, more variety of approximations of trained networks is obtained by changing the initial setup of neural networks. But it is often difficult to find a setup, where the trained network has the same quality as a Kriging model.

If the objective has only a few free variables and there are enough members, it is practical to use neural networks with (too) many weights. Then the redundant weights are set to zero by the training algorithm. In this case different initialized neural networks (with many weights) lead to similar approximations, which are the best solutions in the context of the Bayes'theorem.

In high-dimensional design space the number of input knots is the number of variables plus one (bias knot). So the number of weights raises dramatically with the number of knots in the first hidden layer. This means training takes too much time, if the same strategy like in low-dimensional space is followed. How to find an adequate structure is described in [7].

Another problem is its behavior at extrapolation. The predicted values can decrease (or increase) unacceptable, when the distance to the database is too big (ordinary Kriging predicts the mean value of the objective).

Model Selection in the Present Optimization

In the first phase of the presented optimization Kriging models were used. At the beginning of the optimization, it is an important property that the number of effective parameters is reduced, e.g. to prove that the real objective function is linear, more than $n+1$ members are necessary, if there are n design variables. If the number of variables is reduced less members are needed. So reducing the number of effective parameters increases the confidence in the approximation. At the beginning of the optimization these most important parameters were the stagger angles, spline control points of the duct and the blade count. In later iterations of the acceleration process more and more variables are used inside the surrogate models. In the last third of the optimization all variables were optimized with surrogate models.

Due to the fact, that the extrapolation behavior is superior to neural networks and its easy handling (no structure optimization) Kriging is also used in the later phase of the presented optimization.[1]

Only, when the optimization stagnates (with Kriging created members do not improve the Paretofront), new members are created by neural networks. Changing the surrogate models can solve the problem of stagnation during the optimization.

At the end of the present optimization about 1400 members successfully pass through the complete process chain. Hence in every iteration of the acceleration process all members of the database become training members of the Kriging models.

Training of Surrogate Models

The first training begun after creating 60 members in the CFD/FEM optimization. For each objective several surrogate models are trained, because in high dimensional parameter space the optimization database is not big enough to represent the real objective functions with all their local minimas.

The effect to surrogate models is that different initializations of model parameters often lead to different approximations of the objectives. Hence several models are coupled to enhance the estimation, because unreliable models can be compensated.

Both surrogate models, Kriging and neural networks, use probabilistic methods for fitting their parameters. In case of Kriging a maximum likelihood estimation is performed. Neural networks are optimized by using Bayes'theorem; maximizing the posterior probability of the model parameters. One advantage of probabilistic criteria is, that the complete database can be used for training. This is a very important aspect, when there is a sparsely sampled design space. Another advantage is the fact, that these criteria are differentiable. This allows very effective training algorithms like Quasi-Newton methods.

[1] It is possible that a comparable (or even higher) approximation quality can reached with neural networks, if the network structure is optimized. This will be a topic for further research.

Creating a Database of Members

There are different aspects to consider, when creating new members. First of all they should enhance the optimization objectives and hold the constraints. That means a criterion for new members has to couple numbers with different mean values (e.g efficiency, stresses).

On the other hand the created members will become training members of the surrogate models. So they should also improve the surrogate model approximations. This is usually done by new members in sparsely sampled space, where the uncertainty of the surrogate models is big.

Another aspect of the criterion is, that it has to combine several members. The training of surrogate models and the creating of new members requires a lot of computational resources. Hence it is too expensive to create only one member in one iteration (one training plus one optimization).

Expected Improvement ExpImpr (see [8]) is the product of the expected value of the improvement $E[I]$ and of the probability of improvement $P[I]$. If an objective $f(x)$ is estimated by $\hat{y}(x)$, whose distribution is given by a density function $p(\hat{y}(x))$, the ExpImpr at location x (e.g. design variables) to an objective value b (usually the best fitness value) is formulated as:

$$
\begin{aligned}
ExpImpr(x) &= E[I] * P[I] \\
&= (b - E[\hat{y}(x) | \hat{y}(x) < b]) * P(\hat{y}(x) < b) \\
&= \left(b - \frac{1}{P(\hat{y}(x) < b)} \int_{-\infty}^{b} p(\hat{y}(x)) \hat{y}(x) d\hat{y} \right) \\
&\quad * P(\hat{y}(x) < b) \\
&= \left(b - \frac{1}{\int_{-\infty}^{b} p(\hat{y}(x)) d\hat{y}} \int_{-\infty}^{b} p(\hat{y}(x)) \hat{y}(x) d\hat{y} \right) \\
&\quad * \int_{-\infty}^{b} p(\hat{y}(x)) d\hat{y} \\
&= b * \int_{-\infty}^{b} p(\hat{y}(x)) d\hat{y} - \int_{-\infty}^{b} p(\hat{y}(x)) \hat{y}(x) d\hat{y} \\
&= \int_{-\infty}^{b} p(\hat{y}(x)) (b - \hat{y}(x)) d\hat{y}
\end{aligned}
$$

In our appliance $p(\hat{y}(x))$ is the estimated objective distribution of a surrogate model, which is a normal distribution with the predicted value as mean and the uncertainty estimation as variance. If $p(\hat{y}(x))$ is a normal distribution $\phi(\mu(\hat{y}(x)), \sigma(\hat{y}(x)))$, the integral in the formula for ExpImpr can

be solved:

$$ExpImpr(x) = [b-\mu(\hat{y}(x))] * [\Phi(0,1)]_{-\infty}^{\frac{b-\mu(x)}{\sigma(x)}} + \sigma(\hat{y}(x))\,[\phi(0,1)]_{-\infty}^{\frac{b-\mu(x)}{\sigma(x)}}$$

$\Phi(\mu,\sigma)$ is the cumulative distribution of the normal density function $\phi(\mu,\sigma)$.

Using ExpImpr as a measure, new members are created by searching for the x with the maximum value of the ExpImpr.

In practice, new members are generated frequently in sparsely sampled space, because $p(\hat{y}(x))$ is calculated by surrogate models. The deviation of $p(\hat{y}(x))$ rises in that region, e.g. if $\mu(\hat{y}(x)) = b, \forall x$, the best new member is the member with the biggest $\sigma(\hat{y}(x))$. That implies ExpImpr is a compromise between exploration and exploitation. Thus members created by maximizing ExpImpr enhance the surrogate models in regions of the interesting design space.

In a Paretofront optimization ExpImpr has to be extended to multiple objectives.

Multiple Objectives: There are often multiple objectives in the context of optimization. In this case, the expected gain of volume (ExpVolGain) is calculated. This procedure differs to [8], where a (Euclidean) distance to the reference point $\overrightarrow{b}$ is assigned. The volume is defined as $v(\overrightarrow{a}) = \prod_{i=1}^{n} \lambda_i a_i$, where a_i is the length of edge i and $\lambda_i > 0$ a scaling factor. The advantage of a volume compared to a distance is, that for two volumes $v(\overrightarrow{c}) > v(\overrightarrow{b})$, the relation is preserved by changing $\overrightarrow{\lambda}$. There is no scaling problem for the objectives in that measure.

For n objectives $ExpVolGain(x)$ is formulated as:

$$\begin{aligned} ExpVolGain(x) &= \prod_{i=1}^{n} E_i[I] \\ &\quad * P(\hat{y}_1(x) < b_1 \cap \ldots \cap \hat{y}_n(x) < b_n) \\ &= \prod_{i=1}^{n} (b_i - \frac{1}{\int_{-\infty}^{b_i} p(\overrightarrow{\hat{y}}(x)) d\hat{y}_i} * \\ &\quad \int \cdots \int_{-\infty}^{b_n} p(\overrightarrow{\hat{y}}(x)) \hat{y}_i(x) d\hat{y}_n \ldots d\hat{y}_1) \\ &\quad * \int \cdots \int_{-\infty}^{b_n} p(\overrightarrow{\hat{y}}(x)) d\hat{y}_n \ldots d\hat{y}_1 \end{aligned}$$

$p(\vec{\hat{y}}(x))$ is now a multivariate density distribution of the objectives and $\vec{\hat{y}}(x)$ is a family of the objective functions.

In the DLR optimization framework AutoOpti surrogate models are used for each objective. Hence $p(\vec{\hat{y}}(x))$ is a multivariate normal distribution. By using a diagonal matrix in $p(\vec{\hat{y}}(x))$ as correlation matrix, the assumption that objectives are independent is postulated. Generally, this is not the case (e.g in a two objectives optimization a negative correlation in the *optimized* Paretofront is observed)[2]. On the assumption of independence the formula of ExpVolGain is reduced to:

$$\begin{aligned} ExpVolGain(x) &= \prod_{i=1}^{n} E_i[I] * P_i[I] \\ &= \prod_{i=1}^{n} \int_{-\infty}^{b_i} p_i(\hat{y}_i(x))(b_i - \hat{y}_i(x)) d\hat{y}_i \\ &= \prod_{i=1}^{n} ExpImpr_i \end{aligned}$$

The ExpVolGain is calculated with respect to a member with objective values $\overrightarrow{b}$. In an optimization the ExpVolGain has to be calculated with respect to a Paretofront.

Paretofront Enhancement: To calculate the ExpImpr and ExpVolGain criteria, it is necessary to solve integrals. Hence it is reasonable to divide the space of improvement into disjoint rectangles (see figure 5) or n-dimensional cuboids (see figure 6 for three dimensional cuboids).

Regarding one n-dimensional cuboid cub_k with limits $\overrightarrow{a}$ and $\overrightarrow{b}$ $(a_i < b_i, i \in [1,\ldots,n])$ and independent objectives, the Paretofront enhancement (ParetoEn) in this cuboid is calculated by:

$$\begin{aligned} ParetoEn_{cub_k}(x) &= ExpVolGain_{cub_k}(x) \\ &\quad + \sum_{\psi \in \Psi} \prod_{j \in \psi_{dom}} (b_j - a_j) * \int_{-\infty}^{a_j} p_j(\hat{y}_j(x)) d\hat{y}_j \\ &\quad * \prod_{j \in \psi_{cub_k}} ExpImpr_{cub_k,j} \end{aligned}$$

In the first part the ExpVolGain is calculated, if $\overrightarrow{\hat{y}}$ (the member) is inside the cuboid. The second part treats the case, when the cuboid is dominated by one or more objectives of $\overrightarrow{\hat{y}}$. The lower limits a_j are usually not $-\infty$. Hence parts of the cuboid can be dominated (see figure 8). That space of dominance Ψ is for itself separated into disjoint cuboids, which is the reason for the sum $\sum_{\psi \in \Psi}$. For these cuboids ψ a case differentiation is performed. If an interval is dominating the cuboid cub_k in objective j, the expected value of the improvement is $b_j - a_j$

[2] In our experience the error seems to be usually acceptable, but this is a further research topic

in cuboid cub_k and the probability is calculated for $P(\hat{y_j} < a_j)$. Otherwise the ExpImpr, as previous defined, is applied.

The overall Paretofront enhancement is the sum of the expected volume gain of all L cuboids.

$$ParetoEn(x) = \sum_{k=1}^{L} ParetoEn_{cub_k}(x)$$

Having two objectives, there is only one dominating cuboid ψ for each cub_k, because one a_j is always $-\infty$ in every rectangle. So this objective can not be dominated. But if the number of objective increases, the number of dominating cuboids ψ raises rapidly. Due to this aspect and the also increasing number of cuboids cub_k, the ParetoEn criterion requires a large amount of computational resources in multi-dimensional (objective) space.

The ParetoEn criterion of a member is always positive, if there is the possibility of improvement. This is not necessary the case, if a distance to the Paretofront is calculated as quality criterion for a new member. Because it can be found (see [8]), that the member for the distance calculation is created by averaging the expected objective values of every cuboids. But the space of improvement is not necessarily convex. So this member, a linear combination of the expected values of every cuboid, may not be in this space.

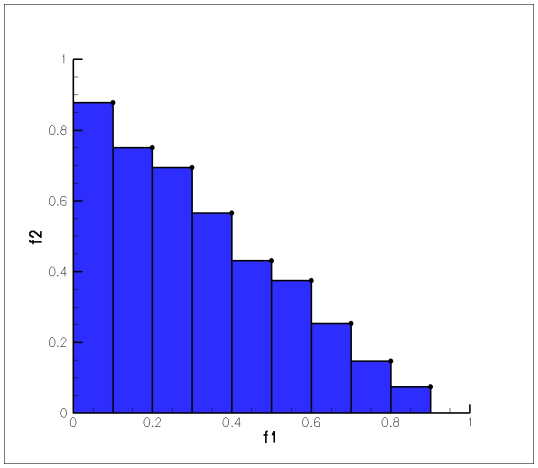

Figure 5. Rectangles of Paretofront dominating space.

Set of Members: When creating a set of l members, l objective distributions have to be included in ParetoEn (figure 7). Calculating ParetoEn can be very time consuming in this case[3].

[3]In a naive algorithm, it is similar to having l objectives

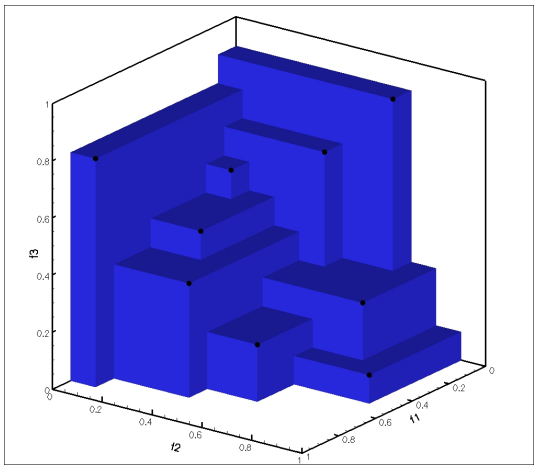

Figure 6. A dominant member has to be inside the block structure.

So, instead of solving the equations analytically, the objective distributions are simulated by creating random samples (figure 8). Thus no integrals have to be solved.

The probability of improvement $P(I)$ is the number of samples in Paretorank 1 space χ divided by the total number of samples.

$$P(I) = \frac{\#\{samples \in \chi\}}{\#\{samples\}}$$

The Paretofront enhancement of one set of members $ParetoEnSet(\kappa)$ is calculated for one random set κ, composed of l random samples (each of the objectives distribution of another member). First the ParetoEn is determined for one sample. If the sample has Paretorank 1, it is added to the Paretofront. Then ParetoEn is calculated for one of the others $l-1$ samples, but relevant to the new Paretofront. The same procedure is repeated for the rest of the samples. So $ParetoEnSet(\kappa)$ is:

$$ParetoEnSet(\kappa) = \sum_{k=1}^{l} ParetoEn(sample_k, Paretofront_k)$$

After determining $ParetoEnSet(\kappa)$, the Paretofront is reset to the initial shape.

The overall Paretofront enhancement $ParetoEnSet(\Omega)$, $\Omega = \{\cup\kappa\}$ of the l objective distributions is performed by the product of the averaged N random sets and the probability of improvement $P(I)$.

$$ParetoEnSet(\Omega) = \frac{1}{N}\sum_{i=1}^{N} ParetoEnSet(\kappa_i) * P(I)$$

When optimizing $ParetoEnSet(\Omega)$ for a set of members, it is important that $ParetoEnSet(\Omega)$ is not changing randomly. Thus samples of the standard normal distribution are created at the beginning of the optimization. During the optimization the objectives distribution of a member is simulated by these samples. The samples are converted to the objectives distribution by scaling and shifting with the deviations and expectations of the member objectives distribution.

Another important aspect has to be considered, when having several times the same member in one set. In this case the distributions are the same (respectively absolutely correlated). Thus in a random set κ the same sample of the standard normal distribution has to be chosen for identical members. Otherwise, it is possible to increase the value of $ParetoEnSet(\Omega)$ by adding a member multiple times in one set of members.

In AutoOpti absolutely correlated distributions are assumed. Thus the members in a set are more spreaded, because similar objective distributions can not improve ParetoEnSet in the same way like uncorrelated distributions. In addition computational time is saved, because otherwise the correlations have to be estimated. This estimation can be performed by surrogate models, whose building process is time consuming.

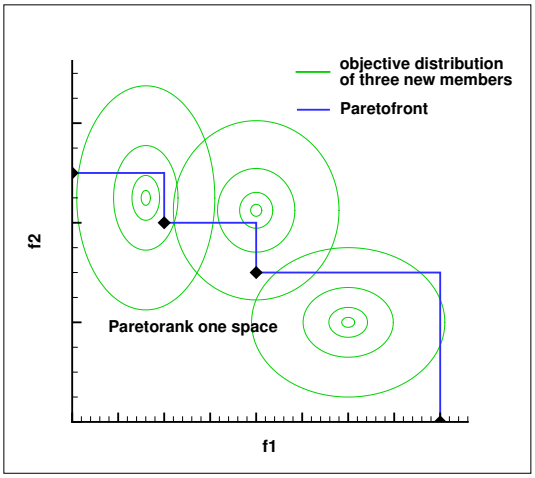

Figure 7. Objective distribution of three members compared to the Paretofront

Handling Constraints: Constraints are easy to combine with $ParetoEnSet(\Omega)$. For a set of members the probability of fulfilling the constraints $P(C)$ is calculated. Then the quality measure for a set of members is the product of $ParetoEnSet(\Omega)$ and $P(C)$. $P(C)$ has a similar meaning as the probability of improvement $P(I)$, which describes the probability of being in the space of interest. Building the product is reasonable, if the constraints and objectives are independent.

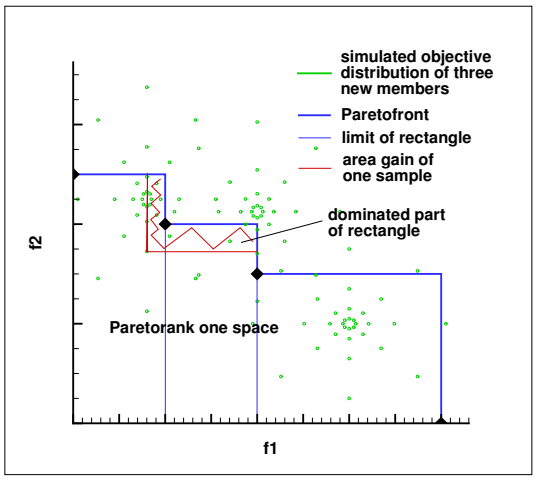

Figure 8. Simulating objective distributions by sampling

Optimization with Surrogate Models: ParetoEnSet is a suitable measure for new members and the surrogate models help to save time in the process chain. Nevertheless it is not practical to optimize until the best possible ParetoEnSet is found, because there are some aspects to regard.

When performing a high-dimensional multi-disciplinary optimization, we always start with a pre-optimized member (in the present optimization with a 3D optimized member). The first new members are created in a surrounding area. Hence the surrogate models are trained with members, which are not representative for the whole design space. Usually, nearly all members in the design space have a worse performance than the initial member. But the surrogate models have no information about this issue. So they will explore the design space, without finding adequate members.

A similar effect emerges, because of the small database compared to the design space. Having a complex objective presented by only a few sampling points (members) results in a simplified approximation of the function.

So the surrogate models maybe not reliable when they used for exploration. To counteract the exploration, a limited number of new members is created. These new members are generated by a genetic algorithm with small step size (e.g. small deviations in mutation). The exploration is additionally restricted, when the objective values are minimized (like in the CFD optimization) instead of maximizing ParetoEnSet directly. In that case ParetoEnSet is used at the end of the optimization for selecting a set of members of the created database.

At this point, it is the task of the designer to modify the parameters (number of new members, parameters of GA (genetic algorithm), minimizing objectives or maximizing ParetoEnSet, etc.). For an optimal result, these parameters are changing during the optimization. A cross validation between the predictions

of the surrogate models and the high fidelity CFD/FEM results helps to find a suitable set of parameters. If the surrogate models are not reliable, the number of created members in the acceleration process is reduced.

In the present optimization it was adequate to create a database of a few hundred members in every iteration of the acceleration process. The ParetoEnSet criterion is only used to select members for the CFD/FEM optimization.

Results

Optimization has been stopped after 1470 successfully evaluated members. 90% of the computational effort was spent for the CFD/FEM calculations and 10% for the surrogate model acceleration process. Figure 9 compares the compressor performance maps of the baseline design and an optimized configuration. Due to the Pareto optimization strategy using working line efficiency and stall margin as objectives, selection of a final result has been made with focus on maintaining baseline stall margin and maximizing the efficiency. Hence, both compressor designs in figure 9 reach a very similar near stall total pressure ratio. Significant improvement of stage efficiency has been obtained for all rotational speeds with an increase of +1.2% points at 100% speed and more than +1.5% points at 79% speed. In figure 10 the radial distributions of the stator loss coefficient are plotted. Compared are the optimized design to the baseline for the two optimization operating points at 100% speed. While the losses are the same (casing) or even higher (hub) in the endwall regions, the profile losses in the core region are significantly reduced. The remarkable reduction of the number of stator blades from 57 (baseline) to 44 contributes to this improvement, while the aerodynamic loading of this already initially very high loaded stator design further increases. This problem is countered with a more sophisticated profiling, improved relative positioning of the two stators and stacking. Figure 11 compares the stator geometries and shows Mach number contours on blade-to-blade planes of 5 and 50 percent relative span for near stall operating conditions at 100 percent rotational speed. High diffusion is achieved in the hub region by the shock and a downstream subsonic deceleration in the first stator row. It is interesting that the "bow" shape of the second stator has been removed for a radially more "stable" flow field. This better controls the radial extension of the blockage area, which results from an interaction of the wake with the hub endwall boundary layer within the second stator passage.

Due to consideration of the stator eigenfrequencies as constraints in the optimization the finally optimized stator designs were free of crossings with the first rotor harmonics.

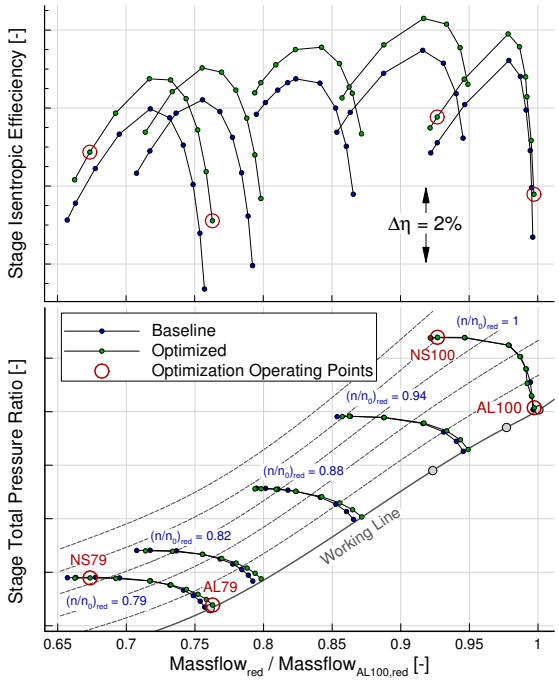

Figure 9. Optimization progress by the compressor performance map

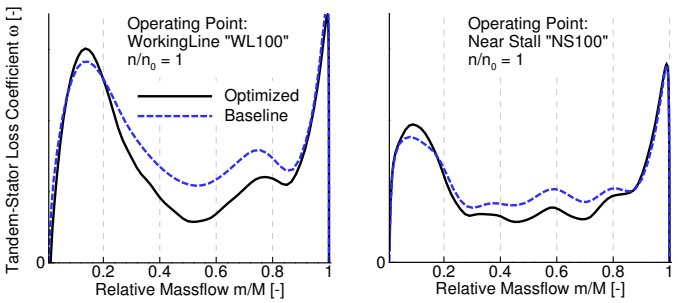

Figure 10. Radial distribution of stator loss coefficient for working line and near stall operating conditions at 100 percent rotational speed

Conclusion and Outlook

A Pareto optimization strategy has been successfully applied to a highly loaded tandem-stator configuration. Within a very limited number of fitness evaluations, compared to the high-dimensional design space, the stator design has been significantly improved both with respect to aerodynamic performance and the stator eigenfrequencies. A computational expensive process chain including four aerodynamic operating points and structural mechanics were handled thanks to the use of surrogate models as acceleration technique combined with a sophisticated selection criterion. The final design passed detailed evaluation of the feasibility of constructional and manufacturing aspects and will be rig tested in 2011.

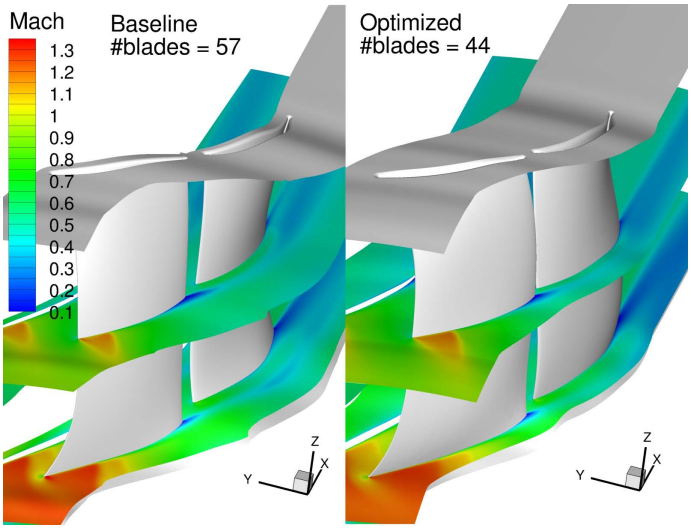

Figure 11. Stator geometries and blade-to-blade Mach number contours in near stall operation at 100 percent rotational speed of the baseline and an optimized design

Future work to accelerate the optimization will concentrate on the surrogate models. At the moment there are some problems to build up the objective functions. Hence we restrict the creation of new members to regions near the samples. For example, it is possible to restrict the extrapolation of Kriging model by limiting the θ_i-values. The exact limits of the θ_i-values can be controlled via cross validation, because during the optimization, new members are permanently created, which were not used for training.

Another option to improve the surrogate models is to use all the information of the CFD-simulations. It is possible to consider a complete flow field solution for a surrogate model. Then the surrogate models are able to predict CFD values for every net point. First tests, by packing the flow field solution with POD (proper orthogonal decomposition), are promising, but very time and memory consuming.

Furthermore an adjoint CFD-program (e.g. Adjoint TRACE) has the ability to improve the surrogate models, because the gradient of an objective can be calculated. So in addition to a single objective value, n values of the gradient are available to enhance the approximation of the surrogate models.

References

[1] Siller, U., Voss, C., and Nicke, E., 2009. "Automated multidisciplinary optimization of a transonic axial compressor". AIAA Aerospace Sciences Meeting 2009, Orlando, USA AIAA2009-863.

[2] Dhondt, G., 2004. The Finite Element Method for Three-Dimensional Thermomechanical Applications. Wiley & Sons.

[3] G. Ashcroft, K. Heitkamp, and E. Kügeler. "High-order accurate implicit Runge-Kutta schemes for the simulation of unsteady flow phenomena in turbomachinery". In V European Conference on Computational Fluid Dynamics ECCOMAS CFD, Lisbon, Portugal, 2010

[4] Kügeler, E., 2004. "Numerisches Verfahren zur genauen Analyse der Kühleffektivität filmgekühlter Turbinenschaufeln". PhD thesis, Ruhr-Universität Bochum, Universitätsbibliothek.

[5] Voss, C., Aulich, M., Kaplan, B., and Nicke, E., 2006. "Automated multiobjective optimization in axial compressor blade design". ASME Turbo Expo 2006, Barcelona, Spain GT2006-90420.

[6] Siller, U., Aulich, M., 2010. "Multidisciplinary 3D-optimization of a fan stage performance map with consideration of the static and dynamic rotor mechanics ". ASME Turbo Expo 2010, Glasgow, UK GT2010-22792.

[7] Schmitz, A., Aulich, M., 2011. "Novel approach for loss and deviation prediction using optimized surrogate models in two-dimensional compressor design ". ASME Turbo Expo 2011, Vancouver, Canada GT2010-45086, in press .

[8] Keane, A. J., 2005. "Statistical Improvement Criteria for Use in Mutliobjective Design Optimization". Automated Design & Optimisation Techniques using CFD 2006, London, UK.

[9] MacKay, David J. C., 1992. "Bayesian Methods for Adaptive Models". thesis.

[10] MacKay, David J. C., 1995."Probale networks and plausible prediction - a review of practical Bayesian methods for supervised neural networks", 1995 - Institute of Physics Publishing

[11] Siller, U., Voß, C., 2010. "Automated Optimization of a Double S-Shaped Inlet for Minimum Loss and Reduced Sight onto the Engine Face", ISROMAC-13 2010, Honolulu, USA, ISROMAC13-TS32.

[12] Lengyel, T., Schmidt, T., Voß, C., Nicke, E., 2009. "Design of a Counter Rotating Fan – An Aircraft Engine Technology to Reduce Noise and CO2-Emissions", ISABE Paper, 19th ISABE Conference Montral/Canada 2009, ISABE 2009-1267.