Representing the Change - Free Form Deformation for Evolutionary Design Optimisation

Stefan Menzel, Bernhard Sendhoff

2008

Preprint:

This is an accepted article published in Evolutionary Computation in Practice. The final authenticated version is available online at: https://doi.org/[DOI not available]

Chapter 4

REPRESENTING THE CHANGE - FREE FORM DEFORMATION FOR EVOLUTIONARY DESIGN OPTIMIZATION

Stefan Menzel[1] and Bernhard Sendhoff[1]
[1]*Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, D-63073 Offenbach/Main, Germany*

Abstract The representation of a design influences the success of any kind of optimization significantly. The perfect trade-off between the number of parameters which define the search space and the achievable design flexibility is very crucial since it influences the convergence speed of the chosen optimization algorithm as well as the possibility to find the design which provides the best performance. Classical methods mostly define the design directly, e.g. via spline surfaces or by representations which are specialized to one design task. In the present chapter, the so-called *deformation methods* are focused which follow a different approach. Instead of describing the shape directly, deformation terms are used to morph an initial design into new ones. This decouples a complex design from an expensive shape description while relying purely on mapping terms which are responsible for the geometry transformations. Thus, the designer is encouraged to determine the optimal relation between parameter set and design flexibility according to the given task. With respect to optimization, these mapping terms are considered as parameters. In this chapter, the combination of two state of the art deformation algorithms with evolutionary optimization is focused. After an introduction of these techniques, a framework for an autonomous design optimization is sketched in more detail. By means of two optimizations, which feature a stator blade of a jet turbine the workability is shown and the advantages of such representations are highlighted.

Keywords design optimization, direct manipulation, evolutionary algorithms, evolutionary optimization, free form deformation, representation, turbine blade

1. INTRODUCTION

Evolutionary algorithms have been successfully applied to a variety of design optimization problems (Sonoda et al., 2004; Oyama et al., 2000; Foli et al., 2006; Ong et al., 2006; Kanazaki et al., 2002). In most cases, the design is represented either by a specialized representation based on standard engineering practice, e.g. circles and connecting curves for two-dimensional blades or an aircraft wing design (Oyama et al., 2000), or by splines, NURBS (Hasenjäger et al., 2005; Lépine et al., 2001) and D-NURBS (Terzopoulos et al., 1994). Other representations like solid modelling (Requicha, 1980) or solutions to partial differential equations (Ugail et al., 2003) have been used less frequently, especially for practical applications.

While standard engineering representations are often very compact, they are almost always incomplete in the sense that not all possible shapes can be represented. Although NURBS are only complete if the representation is adaptive (Olhofer et al., 2001), their versatility is usually much higher. At the same time, the representation of complex shapes with many edges (like e.g. a whole turbine) requires a NURBS surface with a very large number of control points. However, the number of control points defines the dimension of the search space. If the dimension is much larger than 200 and no subspaces can be easily identified to allow a sequential search, the shape design optimization process is prohibitively time consuming and the convergence may not be achieved.

Free form deformation (FFD) techniques, which we will introduce in the next section, are fundamentally different from other representations in that they do not represent the actual shape but changes to a baseline shape. The baseline shape can be arbitrarily complex. Of course the changes that can be represented might be limited by the complexity of the representation. In this chapter, we will introduce free-form deformation and direct manipulation of free-from deformation in the context of evolutionary shape design. We will apply both methods to the design of a stator blade for a real gas turbine engine. Although the strength of FFD is particularly evident for very complex shapes, turbine blades are a reasonable compromise, because their complexity is high enough to require a NURBS representation with just below 100 control points (Hasenjäger et al., 2005) while it is sufficiently simple to analyze the evolutionary process.

2. AERODYNAMIC OPTIMIZATION USING DEFORMATION TECHNIQUES

In order to realize a fully automated aerodynamic design optimization, the so-called deformation techniques are becoming more widely used as an efficient object representation recently. Introduced in the late 1980's in the field of computer graphics (Sederberg et al., 1986; Coquillart, 1990), these methods hold several advantages for representing geometries, especially if these geometries possess a high degree of complexity. Usually the number of parameters to represent such complex geometries, e.g. only by splines or spline surfaces, is too large to be feasible. When applying free form deformation (FFD), the current state-of-the-art deformation algorithm, the object is embedded within a lattice of control points, which defines the degrees of freedom for the deformation. The parameter set and consequently the difficulty of the optimization problem can be tuned by the number and choice of control points or control point groups. When defining the parameter set of a FFD system we have to find the optimal trade-off between search space dimension and design flexibility, i.e. freedom of variation. However, compared to spline based representations, in the FFD framework the complexity of design variations and not of the initial design is the limiting factor.

In case of problems that need finite element or finite volume methods for design evaluations like in computational fluid dynamics, FFD has another advantage (Perry et al., 2000; Menzel et al., 2005). The fidelity of computational fluid dynamics (CFD) simulations depends to a large degree on the quality of the mesh or grid that is used for the simulation. For complex shapes and structures, mesh generation is a very time consuming process (several days), which more often than not requires manual fine-tuning or resolution of meshing problems. In particular, in the context of population based search methods like evolutionary algorithms, manual mesh generation is not feasible. In the FFD framework, the mesh is deformed just like the design is deformed. Therefore, the mathematical deformation procedure is applied to the design and to the mesh simultaneously. This has the great advantage that mesh generation just has to be done once at the beginning of the optimization for the baseline design. During optimization the mesh is always adapted to the changing design. Of course, the quality of

the deformed mesh has to be controlled during the evolutionary search process. However, for complex shape design, the whole optimization process takes days and weeks and the sporadic analysis of the current mesh quality in parallel to the search process is feasible.

In Section 3, two kinds of deformation methods are briefly introduced: the standard free form deformation (FFD) and the direct manipulation of free form deformations (DMFFD) as one of its extensions. The latter method makes use of object points as direct handles on the geometry to take into account the sensitivity of the FFD method to the initial placement of the control points. In Section 4, both methods are used as representations for an evolutionary design optimization of an aerodynamic shape case study. On the one hand FFD is applied to optimise a three-dimensional blade geometry to show the basic behavior of FFD in optimizations while dealing with complex structures. On the other hand both techniques, FFD and DMFFD, are used as representations in a two dimensional blade optimization to illustrate the differences and advantages of DMFFD over FFD.

3. REPRESENTING DESIGNS WITH DEFORMATION METHODS

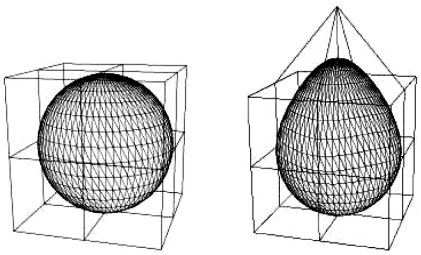
In this section, the methods of free form deformation (FFD) and direct manipulation of free form deformations (DMFFD) are briefly introduced.

3.1 Free Form Deformation (FFD)

The basic idea behind free form deformation is depicted in Figure 4-1 a). The sphere represents the object which is the target of the optimization. It is embedded in a lattice of control points (CP). Firstly, the coordinates of the object have to be mapped to the coordinates in the spline parameter space. If the object is a surface point cloud of the design or a mesh which originates from an aerodynamic computer simulation (as in our example in Section 4), each grid point has to be converted into spline parameter space to allow the deformations. After this process of ‘freezing’, the object can be modified by moving a control point to a new position. The new control point positions

are the inputs for the spline equations and the updated geometry is calculated. Since everything within the control volume is deformed, a grid from computational fluid dynamics that is attached to the shape is also adapted. Hence, the deformation affects not only the shape of the design but also the grid points of the computational mesh, which is needed for the CFD evaluations of the proposed designs. The new shape and the corresponding CFD mesh are generated at the same time without the need for an automated or manual re-meshing procedure. This feature significantly reduces the computational costs and allows a high degree of automation. Thus, by applying FFD the grid point coordinates are changed but the grid structure is kept.

a)



b)

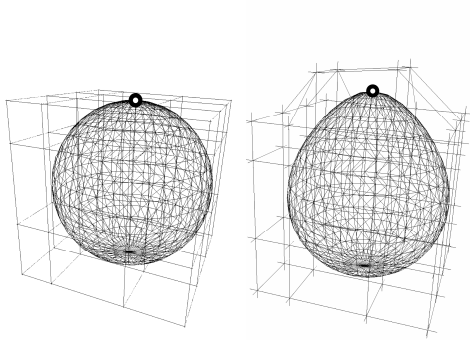


Figure 4-1. Free Form Deformation (Perry et al., 2000). The design is embedded within a lattice of control points. The modification of control points affects the shape as well as everything else inside the control volume. – b) Direct Manipulation of Free Form Deformations. The object point is chosen directly on the surface and the required movements of the control points to realize the target movement of the object point are calculated e.g. by the least squares method. The dotted control volume is invisible to the designer as s/he works directly on the object points; the control volume can be chosen arbitrarily.

As we already mentioned, the number and distribution of control points have to be chosen carefully. As one can imagine, an inappropriate set-up of the FFD control volume increases the necessary size of the parameter set and, therefore, the dimensionality of the search space. One of the reasons is that the impact of a control point on an object decreases when the distance from the object increases. Even a small object variation requires a large modification of the control point if the initial distance between object and control point is large which also violates the strong causality condition that is important in particular for Evolution Strategies (Sendhoff et al., 1997). This in turn often modifies other areas of the design space which has to be compensated for by the movement of other control points. Hence, often correlated mutations of control points are necessary for a local change of the object geometry. To reduce the influence of the initial positions of the control points, DMFFD is considered as a representation for evolutionary optimization. DMFFD allows to determine variations directly on the shape. Therefore, local deformations of the object depend only on the so called object points.

3.2 Direct Manipulation of Free Form Deformations (DMFFD)

Direct manipulation of free form deformations as an extension to the standard FFD has been introduced in (Hsu et al., 1992). Instead of moving control points (CP), whose influence on the shape is not always intuitive, the designer is encouraged to modify the shape directly by specifying so called object points (OP).

Although the initial setup of the control volume is similar to FFD, the control volume becomes invisible to the user and necessary correlated modifications are calculated analytically. In a first step, a lattice of control points has to be constructed and the coordinates of the object and the CFD mesh have to be frozen. But the control volume can be arbitrary, i.e., the number and positions of control points do not need to have any logical relationship to the embedded object, besides the fact that the number of control points determines the degree of freedom of the modifications. In the next step, the designer specifies object points, which define handles of the represented object that can be repositioned. The shape is modified by

directly changing the positions of these object points. The control points are determined analytically so that the shape variations (induced by the object point variations) are realized by the deformations associated with the new control point positions. In other words, the control points are calculated in such a way that the object points meet the given new positions under the constraint of minimal movement of the control points in a least square sense. Of course the object variations must be realizable by the deformations from the calculated new control point positions, i.e., if the number of control points is too small, some variations given by new object point positions might not be representable by a deformation.

In Figure 4-1 b) an object point has been specified at the top of the sphere. The designer is able to move this object point upwards without any knowledge of the "underlying" control volume which can be initialized arbitrarily. The direct manipulation algorithm calculates the corresponding positions of the control points to mimic the targeted object point movement. The solution is shown in Figure 4-1 b).

Direct manipulation of free form deformation has several advantages when combined with evolutionary optimization as compared to standard FFD. First, the construction of the control volume and the number and distribution of control points are not as important as in standard FFD. Furthermore, the number of optimization parameters equals to the number of object points. For an illustration of both methods in the context of an evolutionary design optimization, a stator blade of a jet turbine is considered as a test scenario. The set-up and the results are discussed in Section 4.

4. FFD AND DMFFD IN EVOLUTIONARY DESIGN OPTIMIZATION

In this section, two applications of FFD methods for the evolutionary optimization of aerodynamic structures are described. The aerodynamic problems are well suited to demonstrate both advantages of FFD. They are sufficiently complex and they require CFD calculations for the evaluation.

Evolutionary algorithms belong to the group of stochastic optimization algorithms. They mimic the principles of Neo-Darwinian evolution, see e.g. (Fogel, 1995; Rechenberg, 1994; Schwefel, 1995) by applying operators for

reproduction, mutation and/or recombination and selection. Prominent examples of EAs are Evolution Strategies (ES), Genetic Algorithms (GA) or Genetic Programming (GP). Among the advantages of evolutionary algorithms are robustness against noisy or discontinuous quality functions, the ability to escape from local optima and to enable global search. In the course of optimization, a population of possible solutions (e.g. a vector of continuous parameters, the objective variables) keeps adapted to solve a given problem over several generations. The adaptation occurs by variation of solutions contained in a population and by selection of the best solutions for the next generation.

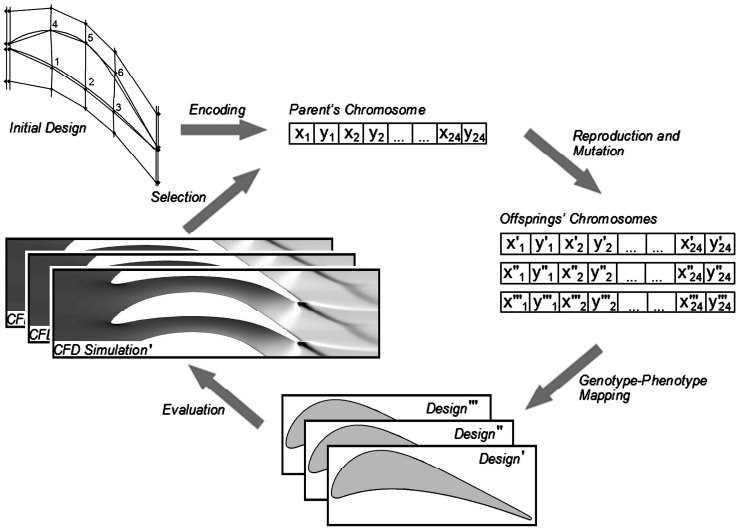


Figure 4-2. The generational cycle in evolutionary design optimization.

Schematically, the evolution cycle is depicted in Figure 4-2, which has already included the turbine blade as the optimization target. In this paper, a special variant of Evolution Strategies, the Covariance Matrix Adaptation (CMA), is applied which has the advantage of a high convergence rate for real-valued problems compared to other evolutionary algorithms. This is particularly important for very time consuming evaluations like CFD simulations. The successful application of this type of algorithm has been

shown previously e.g. for a two-dimensional turbine blade optimization (Sonoda et al., 2004; Olhofer et al., 2001). A detailed description of the CMA algorithm is provided in (Hansen et al., 2001).

4.1 The stator turbine blade as an aerodynamic test scenario

The subject of optimization in this study is a turbine stator blade that is part of a gas turbine for a small business jet. An illustration of the turbine is shown in Figure 4-3. Around the hub of the turbine, eight blades are equally distributed. Because of the low number of stator blades, this design is referred to as an ultra-low-aspect-ratio stator and is less common for the design of gas turbine engines. For more detailed information on the turbine architecture and on the results of a spline-based optimization, the interested reader is referred to (Hasenjäger et al., 2005). Here we will use the design problem of a three dimensional stator blade to highlight the applicability of the standard FFD for complex structures, and a two-dimensional optimization to compare DMFFD with FFD.

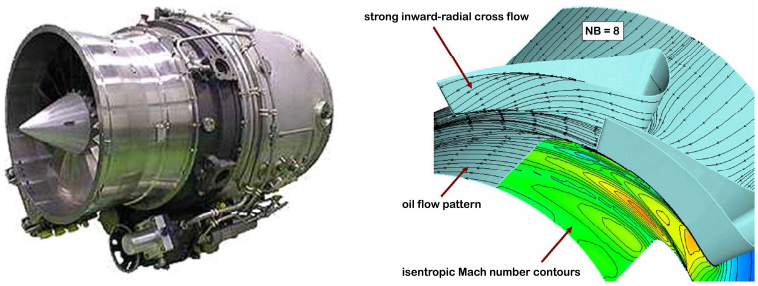


Figure 4-3. Gas Turbine and its fluid dynamics in one blade section (Hasenjäger et al., 2005).

4.2 Evolutionary Optimization of a 3D Structure using FFD

The first step when it comes to numerical design optimization is to extract the characteristic optimization parameters from the representation of the given geometry. In terms of applying free form deformation, as

mentioned above, a lattice of control points has to be constructed which encloses the target geometry. Because of the rotational symmetry in the present problem, only one of the eight turbine blade sections needs to be extracted for further evaluations. In the present problem, the fitness value is calculated via CFD simulation and the region between two blades has to be meshed for solving the numerical equations accordingly. This mesh, which includes the blade's suction and pressure side as boundary layers, was embedded by the lattice of control points to allow the simultaneous deformation of stator blade and computational grid. Finally, twelve control points have been taken from the lattice as optimization parameters. To simplify the calculations and because of the bending of the turbine blade the global x, y and z coordinates of the design and of the knots of the CFD grid have been transferred to a local cylinder coordinate system x', y' and z'. The lattice is fully three dimensional and a sample cross-section is depicted in Figure 4-4. As explained above, the CFD mesh plays an important role in the blade optimization and, therefore, all grid knots that can be found in the CFD mesh between two neighboring blade surfaces have to be fully embedded in the control volume. As a consequence, the deformations are applied to the turbine blade surfaces and simultaneously to the CFD mesh so that a re-meshing process can be omitted.

The blade shape is depicted in Figure 4-4. The continuous upper line is the pressure side and the continuous lower line the suction side of two neighboring turbine blades. The grey region marks the area of the knots and volume cells of the CFD grid. The two blade contours are depicted to show the position of the blades with respect to the control volume. As already mentioned, it should be kept in mind that not the shape of one whole blade is embedded in the control volume but the passage between two blades where the CFD grid is defined. In local x'-direction seven control points have been placed. In local y'-direction the rotational symmetry strongly influences the number and positioning of the control points.

Although all of these control points are important for freezing and deforming the geometry and the CFD mesh, only 12 points have been optimised. Six of these points are shown in Figure 4-2 and Figure 4-4 for the hub section and another six points have been chosen analogously for the casing section. In total 24 parameters (x and y coordinates of the 12 points) have been considered in the evolutionary optimization and were encoded in

the parent's chromosome. To maximize the influence of these control points on the blade geometry they have been positioned as close as possible to the boundary layers of the blade so that the mutation of the control points has a high impact on the design. In this first test scenario due to the small population size the number of optimization parameters has been kept as low as possible. As a consequence, the movements of the 12 control points result in rather global design changes of the blade, which can also be observed from the resulting shapes depicted in Figure 4-6. The present optimization was motivated by the need to correlate these global design changes with the performance differences. A higher degree of locality of the changes can easily be realized by refining the control point lattice by increasing the number of control points, e.g. the optimization with the spline representation discussed in (Hasenjäger et al., 2005) required 88 parameters.

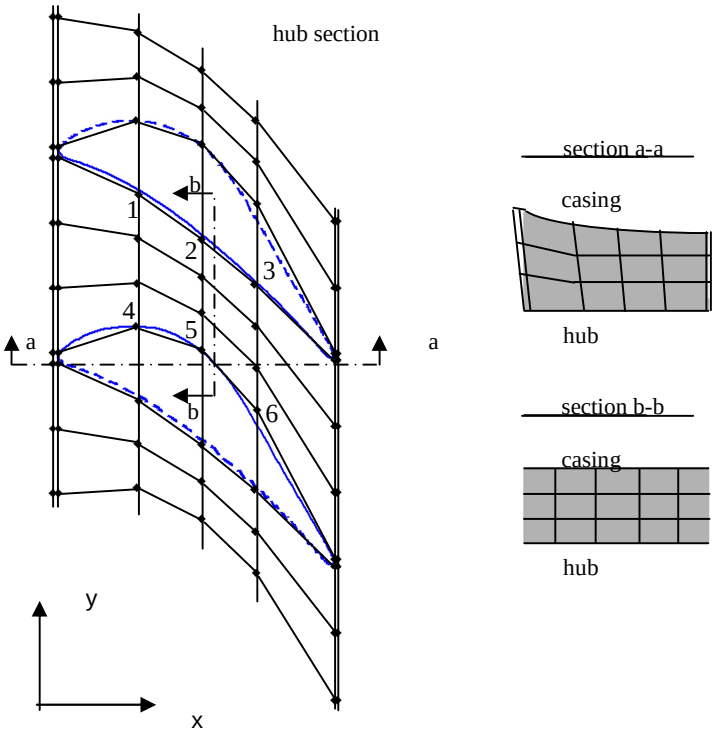


Figure 4-4. Embedding the turbine blade and the CFD mesh in an FFD lattice.

Based on these control point settings the initial CFD grid and the blade geometry have been frozen, i.e. the coordinates of the grid knots have been calculated in spline parameter space. Afterwards the 24 parameters were encoded in the initial parent's chromosome and the optimization was started. The first parameter sets were generated and extracted to calculate the new positions of the control points. Based on these updated control point positions the free form deformation of the CFD grid and the blade geometry were performed and the CFD simulation started.

After the calculation has finished the result of the fitness function is determined by a weighted sum of two flow features and three geometric properties. As the main optimization criterion the minimization of the pressure loss has been chosen. To keep the blade geometry within feasible constraints, four additional values have been extracted from the CFD calculation and blade geometry, respectively. Often an optimum of the fitness landscape is very close to the constraints, hence the boundary conditions have to be checked carefully. To avoid hard constraints, which would directly exclude illegal designs, weights have been introduced so that it was possible to determine a performance index for all evaluations. High penalty terms ($1e^{20}$) have been assigned to these weights w_2 to w_5 which outweigh the contribution of the objective t_1 by far. In case of a violation of constraints, the optimization is quickly driven back to feasible design regions. Before the optimization, the target ranges for the outflow angle, the maximum solidity, the minimum blade thickness and the trailing edge thickness have been defined observing constraints set by other turbine parts, by used materials and by the manufacturing process. Whereas the solidity t_3 of the turbine is a measure for the blade spacing, the blade thickness t_4 and trailing edge thickness t_5 are calculated for a single blade. The calculation is done as follows:

$$f = t_1 + \sum_{i=2}^{5} w_i t_i^2 \rightarrow \min \tag{1}$$

with:

t_1 pressure loss t_2 difference to target outflow angle
t_3 difference to target solidity t_4 difference to target minimum blade thickness
t_5 difference to target minimum trailing edge thickness
w_i weights for the different input data t_2, … t_5

The course of the fitness is depicted in Figure 4-5. Note that the fitness is to be minimized. A total number of 134 generations have been calculated resulting in an overall optimization time of approximately six weeks on a computer cluster. The runtime is closely related to the performance of the CFD solver which depends on the calculation models and computational grid. The overall grid size of one simulation was 175 x 52 x 64 = 582400 cells and the time for the calculation of one blade took about five to six hours on a PIII Xeon, 2.0 Ghz node, depending on the convergence behaviour. As flow solver the parallelized 3D Navier-Stokes flow solver HSTAR3D (Arima et al., 1999) has been used which is perfectly adjusted to the present problem. The solver is parallelized for four CPUs resulting in a total usage of 8 individuals x 4 CPUs = 32 CPUs at the same time. The node communication was realized via the Parallel Virtual Machine (PVM) framework in a master/client configuration (Geist at al., 1995).

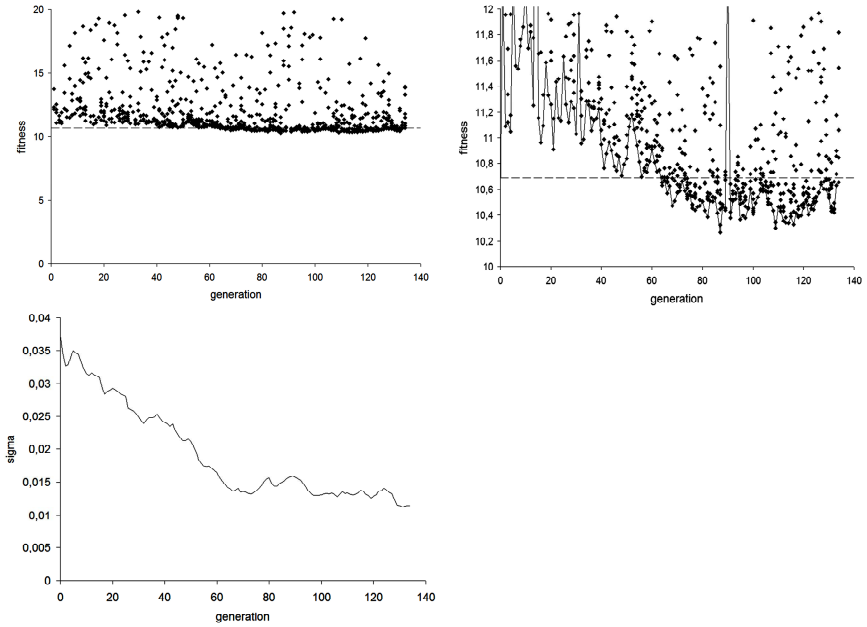


Figure 4-5. Courses of fitness and global step-size during optimization.

In the first ten generations a (1,6)-strategy has been used but was extended to a (1,8)-strategy starting from generation 11 because of the high variance of the fitness values. Generally, a population size above 10 is recommended but could not be realized due to restrictions of the available computer power. The fitness value of the initial blade is about 10.69 and is marked by the dashed line in the fitness graphs.

It can be seen that in the beginning the fitness value of the best design in each generation increases (no elitism) and reaches the initial level again at generation 60. After this point the best individual always performs better than the initial one and stays within a range of 10,27 to 10,60. The best value of 10,27 is reached in generation 87, which corresponds to a performance gain of 4 %.

The course of the optimization can also be analyzed by observing the development of the global step-size. Right from the beginning the step-size decreases and reaches a plateau after approximately 60 generations. At the beginning large mutations were generated leading to an increase of the fitness value. This posed a serious problem due to the small population size of only 6 offspring. Therefore, the population size was increased to 8 individuals starting from generation 11. The initial blade and the shape of the best design from generation 87 are depicted in Figure 4-6 to visualize the changes which occurred.

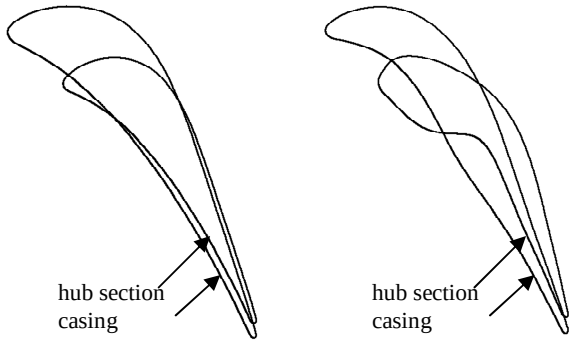


Figure 4-6. Initial and optimized shape of the turbine blade (hub and casing section).

4.3 The Impact of Object Points: A Comparison of FFD and DMFFD as Representations in Evolutionary Design Optimization

As already explained in Section 3, the standard FFD optimization strongly depends on an appropriate set-up of the control volume which relies on the existing know-how of the designer. DMFFD can reduce this influence by introducing object points that can be placed directly on the shape.

In order to compare FFD with DMFFD, we have carried out four optimization runs of a two-dimensional turbine blade (see Figure 4-2). The first optimization uses the standard free form deformation representation and the remaining three the direct manipulation technique. The two dimensional scenario has been chosen because of the large amount of computational resources that are needed for the CFD simulations, especially for the three dimensional flow solver. In all four optimizations, the population size has been set to 32 individuals and an approximation model has been used. In a pre-evaluation step all 32 individuals have been evaluated with a neural network and only the 16 most promising ones have been simulated with the CFD solver to determine the individual fitness. The "true" fitness values have also been used to re-train the neural network online. From the 16 CFD results the best individual has been selected as the parent for the next generation, similar to the standard notation of evolutionary strategies we call this a (1,32(16)) strategy.

Table 4-1. Type and number of parameters.

Run	TYPE	Number of parameters	Number of control points
1	control points	10	10
2	object points	5	10
3	object points	13	10
4	object points	13	36

The details for each run are summarized in Table 4-1. The number of parameters equals the dimension of the search space. Their distribution on the design is depicted in Figure 4-7. The number of control points refers to

the control point coordinates which can be modified in the FFD control volume. This is different from the total number of control point coordinates because points at the upper, lower and left border have to be constant due to CFD mesh consistency. Additionally, control points on the right edge of the control volume can be modified only in y-direction in order to fix the x-length of the design. For run 1 to run 3 the same FFD control mesh is used which is shown in Figure 4-7 in the upper left part for run 1.

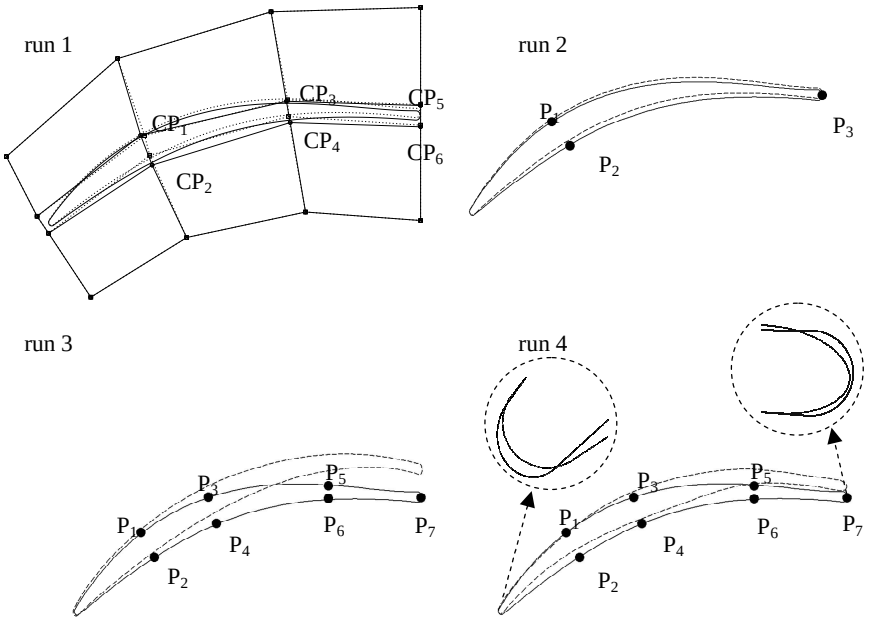


Figure 4-7. Number and distribution of optimization parameters. run1: 10 parameters (P_1-P_4: x, y; P_5, P_6: y); run2: 5 parameters (P_1, P_2: x, y; P_3: y); run3: 13 parameters (P_1-P_6: x, y; P_7: y); run4: 13 parameters (P_1-P_6: x, y; P_7: y). The continuous curve marks the initial designs, the dashed curves the optimized ones. The control volume is only drawn for run 1. The control volumes for run 2 and run 3 are the same as for run 1. Run 4 has been modified in such a way that two rows and columns of control points have been inserted corresponding to a simple knot insertion algorithms as explained in (Piegl et al., 1997).

The general workflow of the design optimization is similar to the optimization described in Section 4.2. A control volume consisting of 4x4 control points has been set up in which the turbine blade is embedded (see

Figure 4-7). For easier visualization the CFD mesh is not plotted. However, we should keep in mind that during the deformation step the blade geometry *as well as* the CFD mesh are modified which allows the omission of the costly re-meshing process.

The control points CP_1-CP_4 can be freely moved in the x-y plane during the optimization, while CP_5 and CP_6 are only allowed to move in the vertical direction as stated above. After the encoding of these parameters (x and y coordinates of points CP_1 to CP_4 and the y-coordinate of CP_5 and CP_6) in the chromosome of the parent individual, the control point positions are optimized. This includes the mutation of the control points, the deformation of the CFD grid based on the free form deformation algorithm and the execution of the CFD flow solver. As described in the previous section, the ES-CMA is used together with a neural network meta-model.

For run 4, the modifications at the leading and trailing edge are shown in a higher resolution to illustrate the occurring deformations. Initial circular or ellipsoid arcs are not kept after deformation because they turn out to be inferior to other leading and trailing edge geometries.

In runs 2, 3 and 4 the direct manipulation of free form deformations is applied to modify the control points directly, i.e.:

1. The chromosomes contain object point positions (P_i) instead of control point positions (CP_i) as parameter sets.

2. The control points are calculated based on the encoded object points with the method for direct manipulation. Here the object points given in Figure 4-7 are used in the three runs. According to Section 3, the procedure of how to calculate the control point positions which are required for deforming the design and the grid is sketched as follows. After the object point positions have been mutated in each generation, the positions of the control points are updated. The new positions of the control points are calculated in such a way that the modifications of the object points are realized as best as possible, i.e. in the present scenario in a least squares sense as described in (Hsu et al., 1992). After an update of the control volume, the design and the CFD grid is deformed and prepared for performance evaluation.

The fitness progressions of all four optimization runs are summarized in Figure 4-8.

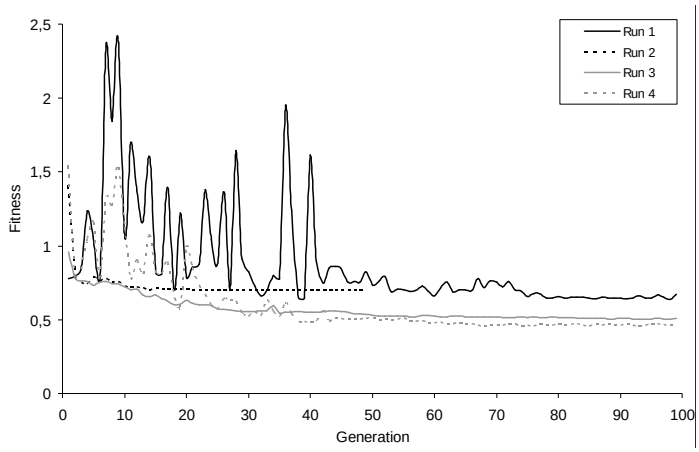


Figure 4-8. The progress of the fitness of the runs 1 – 4.

One major drawback of the direct manipulation method is that the calculation of the control points has to be carried out based on the desired object point positions without being able to incorporate constraints imposed by the CFD grid. In particular, negative volumes can emerge which can be described as loops in the design space.

This can usually be avoided by keeping the order of control points during the deformation step. However, when using direct manipulation a desired object point position can sometimes only be realized by a large degree of control point modifications including destroying the order of control points. Methods for repairing and improving the structure of control points are therefore topic of our current research, cf. (Bihrer, 2006). To guarantee valid CFD meshes, in the present optimization the order of control points is checked after every mutation step. If the order of the control points is changed the mutation is repeated until a valid individual is generated. Figure 4-8 summarizes the results of the optimizations. Base run 1 that uses the standard FFD representation resulted in a converged fitness of 0.62 which means a 37% gain compared to the fitness of the initial turbine blade of 0.98.

According to Figure 4-7, three object points have been chosen for run 2. It resulted in a fitness of 0.7 but it needed less than half the number of generations and the optimization run is very stable. This is due to the reduced number of parameters which is only 5 (2 object points movable in x- and y-, one object point movable in y-direction). However, it also shows that the flexibility of the design is limited by the choice of object points. This demonstrates that an optimization using direct manipulation is limited by two factors. On the one hand, a low number of object points restricts the flexibility of the design because these are the parameters which are optimized. On the other hand, the number of control points limits the degree of realizable shape variations because the control points *actually* induce the targeted object point modifications through the defined deformations. If the number of control points is too low, the targeted object point movements cannot be achieved.

In run 3, the number of object points (OP) has been increased to 7, i.e. 13 optimization parameters (6 OP movable in x- and y-, one OP movable in y-direction) to improve the flexibility of the design. The fitness decreased to 0.5. This is an improvement compared to the optimization run 1. This improvement is particularly interesting because the optimization is based on the same control point grid as in run 1. Even if the number of parameters for the optimization is larger than in run 3, the parameters for the deformations are identical because they are limited by the control point grid. Therefore, the actual design freedom is the same in both runs. Since the number and distribution of control points did not change between both runs the optimization of run 1 must have converged to a local optimum. The structure of the search space seems to be changed by the direct manipulation in a way that the local maximum is avoided in this optimization run. Of course we must be careful to draw conclusions from just one optimization run. Nevertheless, we can state that the different representation of the same degree of variability will lead in general to a different search space behavior.

As a consequence, for this optimization it can be seen that the usage of object points has been more successful. The fitness decreased faster and also at an earlier generation which is particularly important when dealing with time consuming evaluation functions like CFD simulations.

To analyze whether the performance could be even more increased by allowing more flexibility in the possible deformations, two rows and

columns of control points have been inserted into the control volume, resulting in 36 control points in run 4 while the number of object points was kept at 7. The fitness improvement due to the control point insertion is only slightly to 0.45. This is also a promising observation because the number of optimization parameters is still 13 and the course of the fitness is quite similar to the one of run 3. Hence, the increase of flexibility by control point insertion did not affect the convergence behavior.

In summary, we have shown that the usage of the direct manipulation of free form deformation method has been advantageous in many ways in this optimization. If only 3 object points are chosen, like in run 2, the convergence speed improved drastically and resulted still in a good performance compared to the optimization of the control points in run 1. This can be explained by the lower number of parameters in the optimization. If the number of object points is increased, like in run 3, and at the same time keeping the control points fixed, the fitness can be further improved although the possible transformations are kept constant in all three experiments. Here obviously the re-structuring of the search space by the introduction of the direct manipulation methods is beneficial.

Even an increase of control points in the control volume, as it has been done in run 4, did not slow down the optimization. This is a very promising result since the influence of the number of control points did not affect the convergence speed but the number of object points did. As a consequence one could argue for choosing a high number of control points in the optimization to achieve a high flexibility of the transformation and less constraints for the modification due to restrictions in the transformation. This definitely decreases the effect of the control point position and reduces the necessary prior knowledge about the optimization problem while setting up the control volume.

5. SUMMARY AND CONCLUSIONS

In this chapter, the features and advantages of the application of deformation techniques as a representation in evolutionary design optimization have been presented. Even complex designs like automobile parts can be encoded by free form deformation techniques. The limiting

factor is not the complexity of the baseline design but the allowed variations. The fewer the control points, the more global the induced changes. If the control points are positioned based on expert knowledge, even meaningful large scale changes could be represented. At the same time, if a large number of control points is available, local variations can also be realized just as with standard spline representations. For the future, one could envision hierarchical FFD representations which can incorporate (and rely on) expert knowledge to a different degree.

The second decisive advantage of FFD representations for evolutionary design optimization is that the computational mesh for the CFD calculations is deformed together with the design shape. Therefore, a costly re-meshing procedure can be avoided. Indeed for some very complex geometries, re-meshing during optimization is not possible and in turn optimization is only possible with FFD representations. Of course it is still necessary to check that the mesh remains to be feasible after a certain number of deformation e.g. every tenth generation.

Besides applying FFD representations to the evolutionary design optimization of a three-dimensional stator blade of a gas turbine engine, to demonstrate the feasibility in a reasonably complex test scenario, we also introduced direct manipulation FFD as an extension. In DMFFD, design changes are only indirectly encoded in the chromosome. The evolutionary optimization acts directly on object points, however, this only leads to "desired" design variations. These "desired" changes are then realized as close as possible by the underlying FFD algorithm based on a certain number of control points using e.g. a least squares algorithm. Three scenarios are possible. (1) The desired degree of freedom is larger than the realizable degree of freedom – thus evolutionary induced changes might not be realized; (2) the desired degree of freedom roughly equals the realizable degree of freedom – thus most changes can be realized one to one; (3) the desired degree of freedom is smaller than the realizable degree of freedom – thus desired changes can be represented in different ways and therefore, different path' through the search space are available. All three relations are interesting in their own right and deserve a more detailed analysis.

A more practical problem of DMFFD is that constraints for mesh deformation are more difficult to incorporate in the search process. Additional methods for securing mesh consistency must be researched.

Although we have not performed a sufficient number of runs (all design optimization runs with CFD calculations even if meta-models or surrogate models are employed are computationally expensive) to give a clear preference to the DMFFD, it seems that DMFFD will give us more flexibility in the optimization and it will also allow to make design changes more clearly visible to the engineer during the optimization.

References

Arima, T., Sonoda, T., Shirotori, M., Tamura, A., and Kikuchi, K. (1999). A numerical investigation of transonic axial compressor rotor flow using a low-Reynolds-number k-ε turbulence model. *ASME Journal of Turbomachinery*, 121(1), pp. 44-58.

Bihrer, T. (2006). Direct Manipulation of Free-Form Deformation in Evolutionary Optimisation. Diploma Thesis. Computer Science Department/Simulation, Systems Optimization and Robotics Group (SIM), TU Darmstadt, Germany

Coquillart, S. (1990). Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *Computer Graphics*, 24(4):187-196.

Fogel, D.B. (1995). Evolutionary Computation: toward a new philosophy of machine learning New York, NY: IEEE Press.

Foli, K., Okabe, T., Olhofer, M., and Sendhoff, B. (2006). Optimization of Micro Heat Exchanger: CFD, Analytical Approach and Multi-Objective Evolutionary Algorithms. *International Journal of Heat and Mass Transfer*, 49(5-6), pp. 1090-1099.

Geist, A. et al. (1995). PVM: Parallel Virtual Machine – A Users' Guide and Tutorial for Networked Parallel Computing. Cambridge, Ma: The MIT Press.

Hansen, N., and Ostermeier, A. (2001). Completely Derandomized Self-adaptation in Evolution Strategies. In: *Evolutionary Computation*, vol. 9, no. 2, pp. 159-196.

Hasenjäger, M., Sendhoff, B., Sonoda, T., and Arima, T. (2005). Three dimensional evolutionary aerodynamic design optimization with CMA-ES. In: *GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, edited by H.-G.~Beyer et al. ACM Press, New York, NY, pp. 2173-2180.

Hasenjäger, M., Sendhoff, B., Sonoda, T., and Arima, T. (2005). Single and Multi-Objective Approaches to 3D Evolutionary Aerodynamic Design Optimisation. In: *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil.

Hsu, W.M., Hughes, J.F., and Kaufman, H. (1992). Direct Manipulations of Free-Form Deformations. *Computer Graphics*, 26:177-184.

Kanazaki, M., Obayashi, S., Morikawa, M., and Nakahash, K. (2002). Multiobjective Design Optimization of Merging Configuration for an Exhaust Manifold of a Car Engine. In: *Parallel Problem Solving from Nature - PPSN VII*, Guervos, Admidis, Beyer, Fernandez-Villacanas and Schwefel (Eds.), Lecture Notes in Computer Science, 2439, pp. 281-287.

Lépine, J., Guibault, F., Trépanier, J.Y., and Pépin, F. (2001). Optimized Nonuniform Rational B-spline Geometrical Representation for Aerodynamic Design of Wings. *AIAA Journal*, Vol. 39, No. 11.

Menzel, S., Olhofer, M., and Sendhoff, B. (2005). Evolutionary Design Optimisation of Complex Systems integrating Fluent for parallel Flow Evaluation. In: *Proceedings of European Automotive CFD Conference*, pp. 279-289.

Olhofer, M., Jin, Y., and Sendhoff, B. (2001). Adaptive encoding for aerodynamic shape optimization using Evolution Strategies. In: *Congress on Evolutionary Computation (CEC)*. IEEE Press, Seoul, Korea, pp. 576-583.

Ong, Y.S., Nair, P.B., and Lum, K.Y. (2006). Max-Min Surrogate Assisted Evolutionary Algorithm for Robust Aerodynamic Design. *IEEE Transactions on Evolutionary Computation*, 10(4), pp. 392-404.

Oyama, A., Obayashi, S., and Nakamura, T. (2000). Real-coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization. *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*. Springer, Paris, France, pp. 712-721.

Perry, E.C., Benzley, S.E., Landon, M., and Johnson, R. (2000). Shape Optimization of Fluid Flow Systems. In: *Proceedings of ASME FEDSM'00. ASME Fluids Engineering Summer Conference*, Boston.

Piegl, L., and Tiller, W. (1997). The NURBS Book. Springer-Verlag Berlin Heidelberg.

PVM: Parallel Virtual Machine. http://www.csm.ornl.gov/pvm/pvm_home.html.

Rechenberg, I. (1994). Evolutionsstrategie ’94. Friedrich Frommann Verlag · Günther Holzboog, Stuttgart-Bad Cannstatt.

Requicha, A.A.G. (1980). Representations for Rigid Solids: Theory, Methods, and Systems. In: *Computing Surveys* 12, pp. 437–464.

Schwefel, H.P. (1995). Evolution and Optimum Seeking. John Wiley & sons. New York.

Sederberg, T.W., and Parry, S.R. (1986). Free-Form Deformation of Solid Geometric Models. *Computer Graphics*, 20(4):151-160.

Sendhoff, B., Kreuz, M., and von Seelen, W. (1997). A condition for the genotype-phenotype mapping: Causality. In: *Thomas Bäck, editor, Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*, Morgan Kauffmann, pp. 73-80.

Sonoda, T., Yamaguchi, Y., Arima, T., Olhofer, M., Sendhoff, B., and Schreiber, H-A. (2004). Advanced High Turning Compressor Airfoils for Low Reynolds Number Condition, Part 1: Design and Optimization. *Journal of Turbomachinery*, 126: 350-359.

Terzopoulos, D., and Qin, H. (1994). Dynamic NURBS with geometric constraints for interactive sculpting. In: *ACM Transactions on Graphics* 13, Nr. 2, pp. 103–136.

Ugail, H., and Wilson, M.J. (2003). Efficient shape parameterisation for automatic design optimisation using a partial differential equation formulation. In: *Computers & Structures* 81, pp. 2601–2609.