Paper 24545
Aerospace Technology Congress
11-12 October 2016, Solna, Stockholm

An optimization platform for high speed propellers

A. Capitao Patrao*, G. Montero Villar*, J. Takachi Tomita**, C. Bringhenti**, R. Avellan***, A. Lundbladh***, T. Grönstedt*

*Chalmers University of Technology, Göteborg, Sweden
** Instituto Tecnológico de Aeronáutica (ITA), Sao Jose dos Campos, Sao Paulo, Brazil
***GKN Aerospace, Trollhättan, Sweden

ABSTRACT

To improve the efficiency by which current power plants translate jet energy into useful thrust the use of turboprop and in particular open rotor aircraft are being revisited. One challenge in association with developing new powerplants for such aircraft is high speed propeller design in general and noise prediction in particular.

The Boxprop was invented in 2009 by GKN Aerospace in order to mitigate the effects of the tip vortex on noise and to improve upon the aerodynamics of a conventional propeller blade. The Boxprop is composed of a double-bladed propeller joined at the tips, and the design has the potential to eliminate the tip vortex, and thereby decrease that particular noise source. The complex and highly three-dimensional shape of an advanced propeller blade is challenging to model with classical propeller design methods, requiring instead more sophisticated optimization methods.

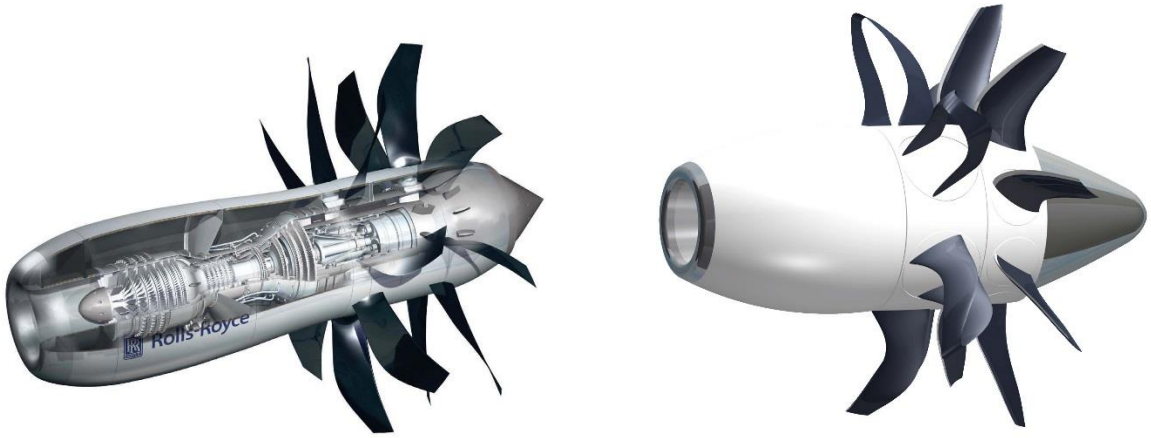
This paper presents an optimization platform developed for high speed propellers, and illustrates its use by performing a reduced aerodynamic optimization of the Boxprop. The optimization process starts by performing a Latin Hypercube Sampling of the design space, and analyzes the resulting geometries using CFD. A meta-model employing radial basis functions is then used to interpolate on the obtained CFD results, which the GA uses to find optimal candidates along the obtained Pareto front. These designs are then evaluated using CFD, and their data added to the meta-model. The process iterates until the meta-model converges.

The results of this paper demonstrate the capability of the presented optimization platform, and applying it on the Boxprop has resulted in valuable design improvements and insights. The obtained designs show less blade interference, more efficiently loaded blades, and less produced swirl. The methodology for geometry generation, meshing and optimizing is fast, robust, and readily extendable to other types of optimization problems, and paves the way for future collaborative research in the area of turbomachinery.

INTRODUCTION

Since its inception, air travel has steadily increased year by year. From 1972 and onward, the annual increase in total passenger kilometers travelled and fuel consumption has averaged 5.8% and 2.2%, respectively [1]. These trends are not likely to change anytime soon, but economic, environmental, political, and consumer pressure is forcing the airplane and engine manufacturers to develop more innovative approaches aimed at decreasing emissions and fuel consumption from air travel.

A particularly interesting concept, offering a potential fuel reduction of 20-35% compared to current engines [2], is the *counter-rotating open rotor* jet engine (CROR), exemplified in Figure 1a. The main fuel-saving feature of the CROR is the two rows of counter-rotating propeller blades situated outside of the nacelle. These blades increase the bypass ratio of the engine, thereby decreasing the fuel consumption. The main drawback of this concept is acoustics - the tip vortices and wakes of the front propeller blades impinge on the rear blades, resulting in higher noise than for conventional turbofan engines. In order to realize the potential fuel-savings, the noise levels need to be alleviated. Schnell [3] reported success in decreasing the noise levels in certain directivity ranges, and Van Zante [4] reported noise levels on advanced CROR designs that are below the noise requirements stated in the ICAO Chapter 4 standard. This has been achieved mainly by clipping the rear rotor of the CROR, thereby preventing the front rotor tip vortices from impinging on the rear rotors, and by optimizing the blade shape for reduced noise. Unfortunately, clipping decreases the propeller disc area, leading to lower efficiency.



a) **b)**

Figure 1 – a) Rolls Royce CROR concept [5], and b) the Boxprop concept used as the front rotor of a CROR [6].

Decreasing the acoustic signature of the CROR is key to its commercialization, especially when considering that future noise regulations will be even more stringent than current regulations (ICAO Chapter 4). A slightly different approach to noise reduction is to employ more radical propeller shapes and configurations, exemplified in this paper by the *Boxprop* (front rotor of Figure 1b), which was introduced in 2009 by Richard Avellan and Anders Lundbladh [7]. The Boxprop can be set up as seen in Figure 1b, with a Boxprop as the front rotor and a conventional propeller as the rear rotor, or in a setup where both the front and rear rotors are Boxprops. The Boxprop consists of pairs of swept blade halves joined at the tips, forming a blade arch as shown in Figure 2. The blade halves are named *Leading Blade* (LB) and *Trailing Blade* (TB) relative to the rotational direction. The potential benefits of the Boxprop lie in its design, the joined blade tips have been shown to weaken the tip vortex [8] present on conventional propeller blades, which

might reduce the acoustic signature. If a Boxprop can be designed to deliver at minimum equivalent performance as a conventional propeller blade, and in combination with a reduced acoustic signature, then the aircraft engine industry will be one step closer to the introduction of the CROR into the general market. Thus far, research has shown [8] that an isolated, single Boxprop (not in a counter-rotating setup) can reach the necessary thrust required if it is to replace the front rotor of a CROR, albeit with more downstream swirl than conventional propeller blades. Consequently, there is a need for optimizing the Boxprop from an aerodynamic point of view.

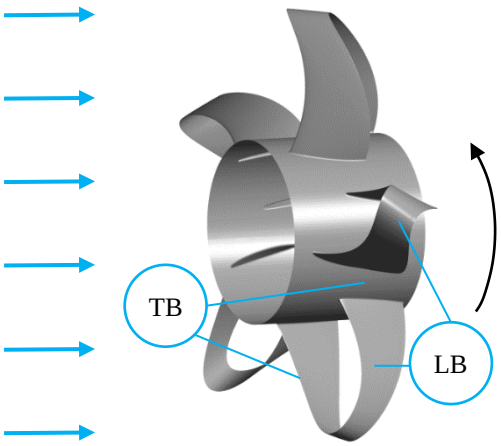


Figure 2 – GPX701 Boxprop with direction of airflow and rotation. Leading (LB) and trailing blades (TB) relative to the direction of rotation are marked.

The aim of this paper is to present a *multi-objective optimization platform* developed for the optimization of high speed propellers. The complex and highly three-dimensional shape of modern high speed propellers is challenging to model with classical propeller design methods, requiring instead more sophisticated optimization methods coupled with CFD. Previous work at Chalmers and ITA has demonstrated the usefulness and effectiveness of these types of approaches, and in particular *Genetic Algorithms* (GA). These algorithms have been used extensively and effectively in the optimization of transonic compressors ([9], [10] and [11]), multistage axial compressors with variable geometry [12], and in the design process of axial turbines ([13] and [14]). For CRORs, Schnell [3] showcased interesting performance and noise results for a similar optimization approach, in which 1600 individual geometries were simulated. Considering the existing in-house experience with GAs and the successful application of it in CRORs by Schnell, a GA based on the *NSGA-II algorithm* [15] has been implemented, which is coupled with a *radial basis function* (RBF) *meta-model*. A number of sample studies were done to exemplify the optimization platform, the foremost being an optimization study of the Boxprop, which is presented here. This multi-objective optimization maximized efficiency and thrust, and associated results will be presented to illustrate the optimization platform.

METHODOLOGY

The optimization platform consists of the different components shown in Figure 3. The geometric parametrization of the propeller blades spans a multi-dimensional design space, which is sampled using a *Latin Hypercube Sampling* technique, resulting in an initial design set of blade geometries. The blade geometries describing the initial design set are then created, meshed, and analyzed using CFD. The values of the objective functions (e.g. thrust and efficiency) are evaluated for each design, and a meta-model employing radial basis functions (RBF) is used to approximate the objective functions as a function of the design set parameters. The GA is then used to find optimal candidates based on the values of the objective functions provided by meta-model. The designs of the optimal candidates are then meshed, evaluated using CFD, and their performance data is added to the meta-model. This process iterates until the meta-model converges and a final Pareto front of optimal candidates is obtained.

The parts of the optimization platform are all coded in *Python* except for the meshing software *ICEM CFD* and the CFD solver *ANSYS CFX*. The subroutines handling the initial design set, geometry generation, CFD, meta-model, and GA are all integrated into one main Python script which manages the entire optimization process. The individual parts of the optimization platform will be described in more detail in the following sections.

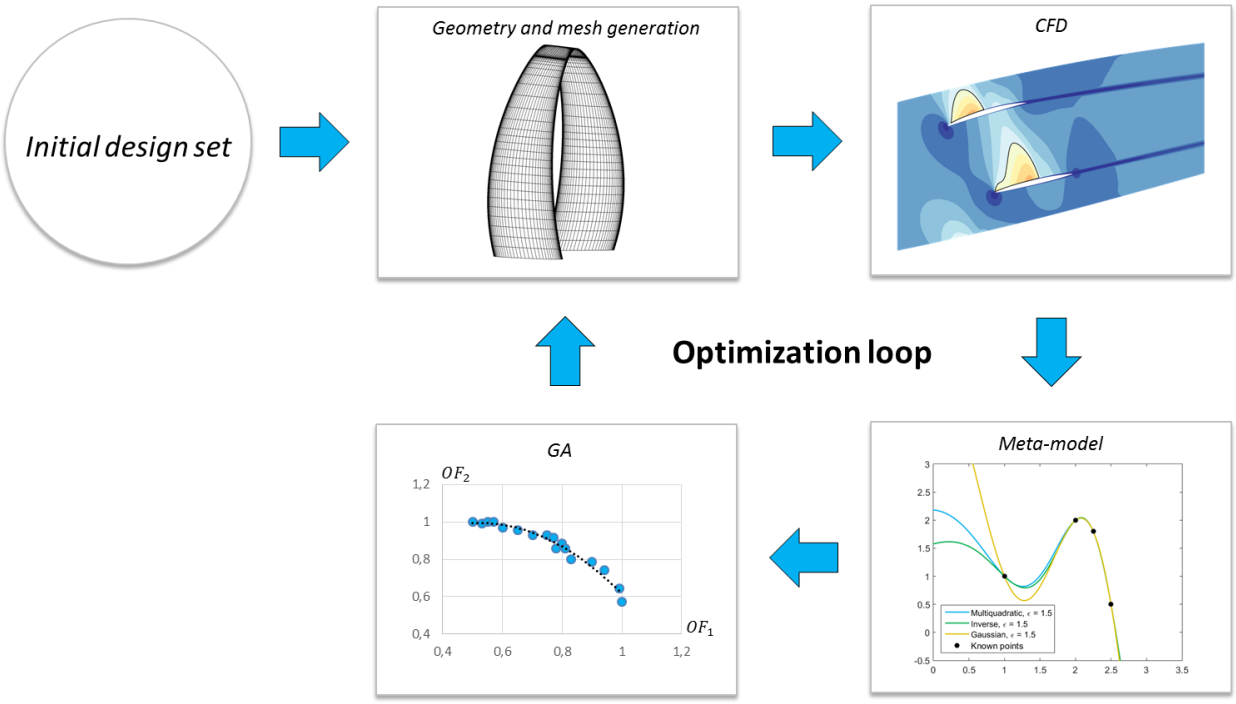


Figure 3 – Components and workflow of the optimization platform.

Blade parametrization

The blade parametrization has been developed with *two main objectives* in mind:

1. *Individual adjustment* of blade angles and camber for each blade half. Previous work [16] has indicated that in order to reach the target blade sectional loading, the blade angles and camber of the LB and TB have to be independently adjusted, taking into consideration the effect of induced flow on each blade half.
2. The *spacing between each blade half* has a large effect on their loading ability. It has been observed in previous work ([8], [16]) that a low pressure region (blade interference) forms in the blade passage between the blade halves when the axial and tangential spacing is too small, effectively decreasing the thrust that can be generated in the TB near the tip. The parametrization should aim for the ability to control both the axial and tangential spacing between each blade half, thereby allowing more thrust to be generated closer to the tip.

The Boxprop blade is built by placing *NACA-16* airfoils along a stacking line (see Figure 4), which has been defined with two cubic *Bézier curves*, one for each blade half, which are required to be tangential to each other at the blade tip. The parametrization of the stacking line includes a pair of hub points which define the location of the blades at the hub radius, and three control points at the blade tip, which define tip position and the shape of the stacking line near the tip. Additionally, a control point for each blade half is added at a radius near mid-span, which allows shaping of the stacking line in that region. Only the endpoints of the Bézier curves (at the hub and tip) coincide with the actual stacking line, for the rest of the control points, coincidence is not guaranteed. By using this parametrization of the stacking line, control over the axial and tangential spacing of the blade halves can be achieved.

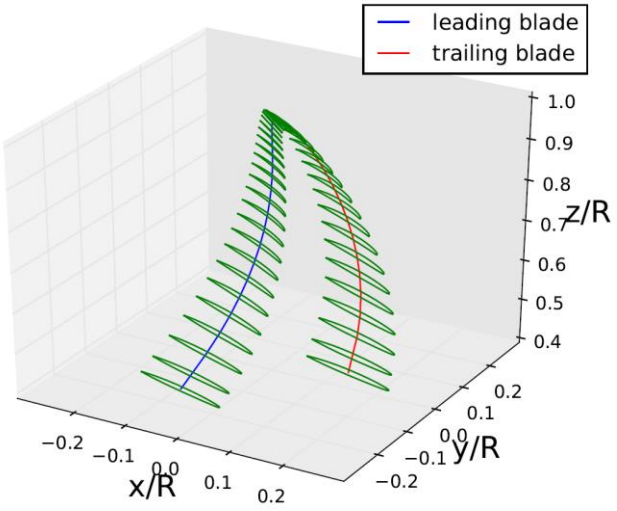


Figure 4 – Boxprop stacking line (blue and red) with stacked airfoil profiles (green). Image credit: [20]

The blade airfoils are defined for each point along the stacking line using blade section distributions of *chord*, *camber*, *thickness*, and *blade angle* as functions of radius. The chord and thickness distributions are quadratic functions while the camber and angle-of-attack distributions are defined using quartic Bézier curves. In this manner, the blade halves can be pitched and cambered individually to reach the desired sectional loading, as was described in the first objective of the blade parametrization.

In total, the Boxprop parametrization amounts to *33 design variables*. For this *reduced demonstration* of the optimization platform, the number of variables has been decreased by excluding the chord and thickness distributions as optimization variables, thereby decreasing the dimensionality of the design space. The camber distribution has also been set as fixed for the purpose of this study, and its radial distribution is similar to the one employed by the NASA SR-7L propeller [17]. An additional simplification of the blade geometry was done by assuming that the control points of the blade stacking line are *odd-symmetric* (see Figure 5b)) with respect to the flow direction and a line going from the origin to the blade tip. With this symmetry condition, only one half of the Boxprop blade needs to be defined, whose stacking line is shown in Figure 5a). The hub control point (P_1) lies on a fixed radius, and its position is set by using the distance d and angle κ. The same approach is used for the control point at mid-span (P_2) and at the blade tip (P_3), with the significant difference that the mid-span radius is an optimization variable, while at the tip the z-coordinate for P_3 is set equal to P_4. The final number of design variables employed for this study added up to 17 variables; 10 variables dictate the angle-of-attack distribution for each blade half, while 7 variables are used to define the stacking line.

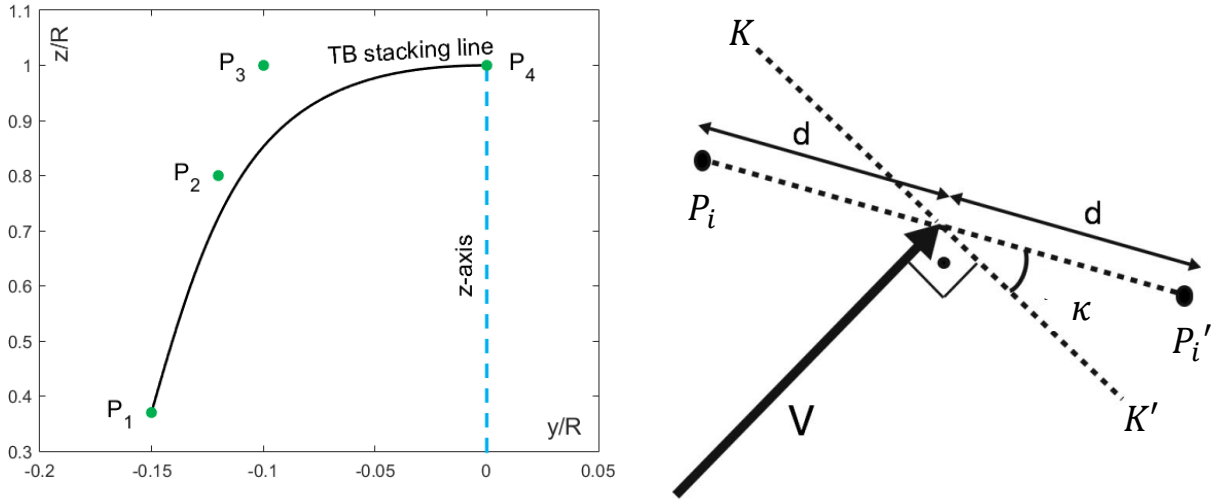


Figure 5 – a) The stacking line of the TB projected onto the YZ-plane (x-direction constitutes the flight direction) together with the control points necessary for its construction. b) Control points P_1 to P_3 at their respective radii are defined using a distance d and angle κ as is shown in the figure for the points P_i and P_i'. The points P_i and P_i' lie along a line which is at an angle to the line K-K'. This line is in turn perpendicular to the flow direction at the chosen radius and the z-axis. Image credit: [20]

Latin Hypercube Sampling

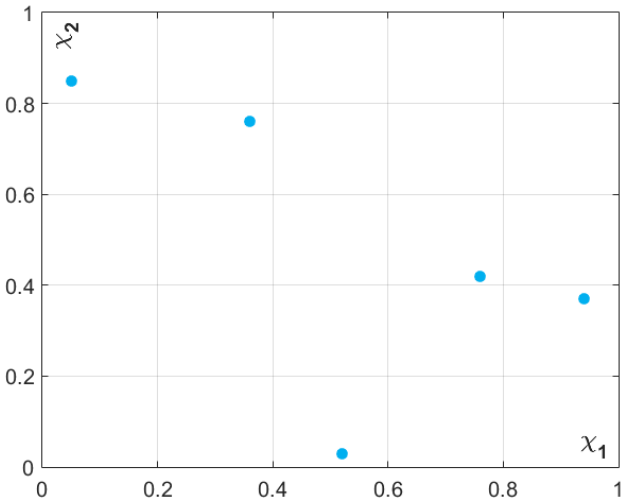


Figure 6 – Five point LHS sampling of the space spanned by variables χ_1 and χ_2. Image credit: [20]

The first process of the optimization platform is to create an initial design set by means of a *Latin Hypercube Sampling* (LHS) technique, exemplified in Figure 6. This technique randomly samples the k-dimensional design space with a pre-determined number of samples N (initial designs). The LHS divides the ranges of the design variables χ_i into N equidistant subranges, and only allows one sample point to occupy each subrange. Additionally, the placement of the sample point inside the subrange has been randomized. In this way, each design variable is well-sampled and the design space in randomly searched. The number of designs used for the initial design set is calculated using Eq. (1), and is the amount of designs required define the k-dimensional, 2nd order polynomial defined in Eq. (2). This study encompasses 17 variables, thereby requiring 171 initial designs in order to define Eq. (2).

$$N = \frac{(k+1)(k+2)}{2} \tag{1}$$

$$\sum_{i,j=1}^{n} \left[A_{ij}\chi_i\chi_j + B_i\chi_i \right] + C = 0 \tag{2}$$

Geometric representation

Previous work with the Boxprop has utilized an in-house MATLAB script for geometry generation, which outputs data for either Autodesk Inventor or CATIA. The CAD software creates the 3D geometry, exports it to a STEP file, which is read by the meshing preprocessing software. This approach works relatively well for a small number of cases, but is unsuitable for an optimization platform. The approach needs considerable manual input from the user, since the creation of the complex, thin 3D blade geometry occasionally fails or needs to be broken up into sections. Beyond being time-consuming due to the need of manual input, the CAD software requires expensive licenses, which is undesirable when running a large number of cases.

The present approach has re-used parts of the existing in-house MATLAB script and created a new *Python* script which creates all of the geometry using open source modules such as *NumPy*, *SciPy*, and *matplotlib*. The output of the geometry script is a *STL file*, a CAD file format extensively used in the area of rapid prototyping and 3D printing. Triangular patches is used to describe the 3D geometry, and the main drawback of this approach is the resulting discontinuous slope and non-existing curvature. Since the computational mesh itself has the same drawback, the only requirement is that the geometry has a sufficiently high resolution in areas with high slope and curvature, such as in the leading and trailing edges of the blade. The developed approach is fast, robust, easy to implement in Python, and offers a high degree of control.

Meshing

The computational domain is subdivided into an *interior 3D cylindrical sector* containing one Boxprop blade and an *outer quasi-2D domain*, see Figure 7. This arrangement allows the pressure field to extend far from the blade without resulting in a very large mesh. This arrangement has been used in previous work [16] and is well-suited for estimating propeller performance such as thrust and efficiency.

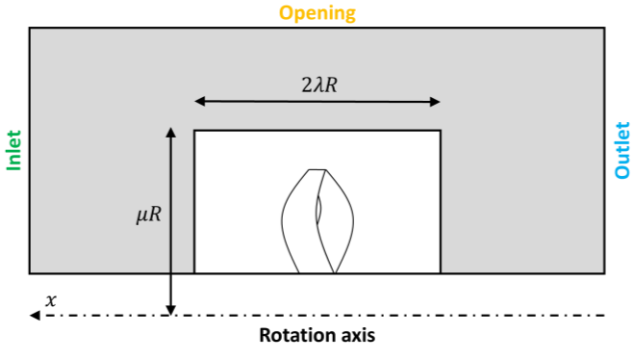


Figure 7 –Outer 2D domain (grey) and inner 3D domain (white). Boundary conditions are marked with Inlet (green), Opening (yellow), and Outlet (blue). Also shown are the domain size parameters μ and λ. Reproduced from: [20]

ICEM CFD is used to create the *structured, hexahedral mesh*, and is automated using ICEM CFD scripts. The STL files of the Boxprop blade are imported together with any additional help surfaces that the meshing requires. The geometry creation mentioned earlier in this paper is also used to create a large amount of help geometry such as points and lines, which the meshing software needs for proper mesh edge and vertex association. Two aspects of the mesh generation are critical, the first being that points and lines of the help geometry must coincide with the edges and points of the STL surfaces, otherwise errors will occur. Secondly, every geometric entity in ICEM CFD has a unique identifier, which requires the user to keep track of every point, line, and surface, otherwise causing ICEM CFD to crash.

A *domain and mesh study* aiming at decreasing the computational cost of the optimization study has been carried out. The size of the inner 3D domain was decreased as much as possible, while the mesh was made coarser in regions outside of the blade O-grid. All cases adhere to an average first node height of $y^+_{ave} = 1$ and an expansion ratio of 1.2 in the O-grid edges normal to the blade surface. Table 1 shows moderate changes in thrust and efficiency for mesh sizes down to 0.74M cells (M5), which was used as a starting guess for the domain study (D2). As is shown in Table 1, the domain size parameters μ and λ (see Figure 7) had little effect on the performance values. A mesh size of 1.15 million cells was therefore chosen as a compromise between computational cost and fidelity.

Table 1 - Mesh and domain study for the Boxprop optimization. All changes are relative to case M1 (bold). Changes in thrust coefficient C_T are proportional, while the absolute difference is shown for the efficiencies.

MESH STUDY

Case	N. cells [10^6]	ΔC_T	$\Delta\eta$[%]
M1	**4.25**	**0%**	**0%**
M2	2.48	0.35%	0%
M3	1.37	−0.74%	−0.40%
M4	0.98	−0.71%	−0.49%
M5	0.74	−1.06%	−0.60%

DOMAIN STUDY

Case	N. cells [10^6]	μ	λ	ΔC_T	$\Delta\eta$[%]
D1	0.74	4	2.4	−0.85%	−0.62%
D2 (M5)	0.74	2.8	1.2	−1.06%	−0.60%
D3	0.74	2	0.6	−0.85%	−0.62%

Modelling of the nacelle boundary layer was omitted since it would bring considerable complications in terms of blocking structure and possibly bad mesh quality for the Boxprop rotating hub surface. Including this in future research is an option, but the effect of the hub boundary on the wake structures is deemed to be small since the boundary layer is considerably smaller than the blade height.

CFD setup

The steady-state CFD simulations necessary for the optimization effort were performed using *ANSYS CFX*, solving for the compressible flow equations using the SST turbulence model, which has been shown in previous work to be appropriate for 3D turbulent flow simulations of turbomachinery components, using either unstructured and structured mesh environments ([18]). A low-Re approach for the modelling of the boundary layer was employed. The 3D domain shown in Figure 7 is set as rotating, and is connected to the outer 2D domain using Frozen Rotor interfaces. Only one blade passage is simulated, and periodic boundary conditions are used to account for the effects of the remaining propeller blades.

The *boundary conditions* are set according to Figure 7 with total pressure, total temperature, and turbulence intensity for the inlet, and static pressure for the outlet. The opening boundary condition is set as an entrainment, in which zero gradient turbulence is set. The hubs are set as free-slip walls, while the propeller blade surface is set as a no-slip surface.

Meta-model

Since the optimization framework employs a GA which is stochastic by design and requires numerous evaluations of the objective functions (thrust and efficiency), there is a need to create an approximation of the objective functions with respect to the design variables χ_i. Otherwise, the computational effort needed to evaluate hundreds of designs with CFD would be prohibitive.

The *Radial Basis Function* (RBF) is a type of interpolation method where the interpolated value $\hat{y}$ for a point $\boldsymbol{x}$ is a function only of its Euclidean distance r_j from the known data points $(\boldsymbol{x}_j, y_j)$ [19]. In this case, $\boldsymbol{x}$ corresponds to a vector of the design variables χ_i, and $\hat{y}$ to an interpolated value of the objective functions:

$$\boldsymbol{x} = [\chi_1 \quad \chi_2 \quad \cdots \quad \chi_n] \qquad \hat{y} = C_T \quad \text{or} \quad \hat{y} = \eta \tag{3}$$

The interpolated values (response surface) obtained from employing a radial basis function are calculated according to Eq. (4).

$$\hat{y}(\boldsymbol{x}) = \sum_{j=1}^{N} w_j \phi(r_j) \qquad r_j = \|\boldsymbol{x} - \boldsymbol{x}_j\| \tag{4}$$

Various types of *bases* $\phi(r_j)$ can be used for the interpolation, examples include *multiquadric*, *inverse*, and *Gaussian*:

$$\phi(r_j) = \sqrt{(r_j/\epsilon)^2 + 1} \qquad \phi(r_j) = 1/\sqrt{(r_j/\epsilon)^2 + 1} \qquad \phi(r_j) = e^{-(r_j/\epsilon)^2} \tag{5}$$

(Multiquadric) *(Inverse)* *(Gaussian)*

The weights w_j in Eq. (4) can be determined by calculating the values of the bases $\phi(r_j)$ for the known data points $(\boldsymbol{x}_j, y_j)$ and solving the resulting linear equation system for w_j.

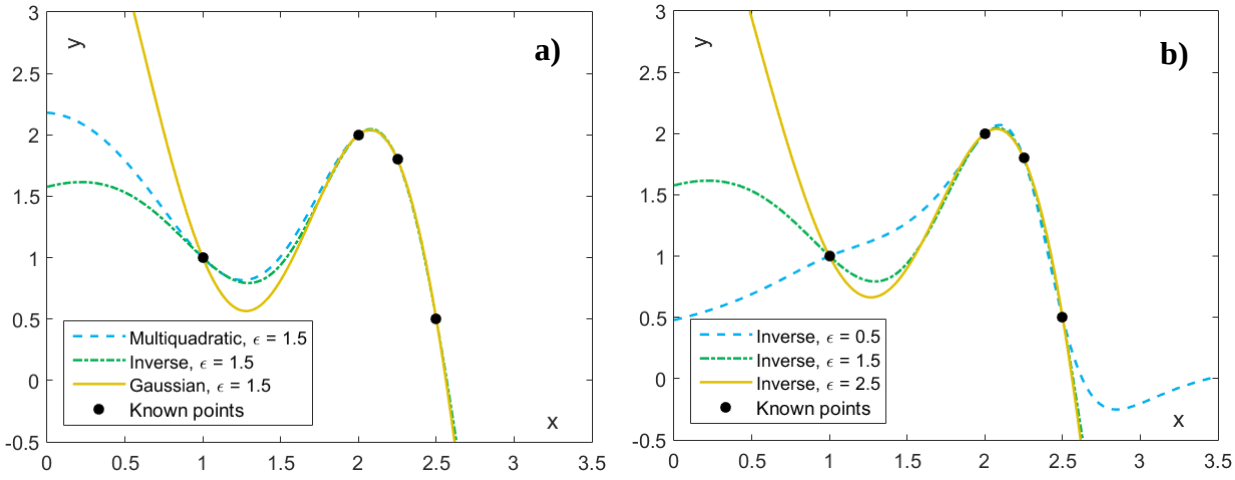


Figure 8 – a) The effect of a) basis $\phi(r)$ and b) ϵ on the response surface of a one-dimensional RBF. Image credit: [20].

The value chosen for ϵ and the type of basis has a great effect on shape the response surface, which is shown in Figure 8. By definition all RBFs intersect the known data points $(\boldsymbol{x}_j, y_j)$, but vary considerably outside of them, leading to very different approximations of the response surface.

In this paper an approach is used that minimizes the error connected to the choice of bases and ϵ:

1. Use most data points for establishing the RBF, but exclude a fraction of them for later.
2. Construct the RBF with the three different bases and with different values of ϵ. Find the appropriate weights w_j for the given data points.
3. Use the obtained RBF's to interpolate values for the known data points that were left out in step 1 and calculate the errors for each combination of bases and ϵ.
4. Choose the most suitable combination of bases and ϵ for each objective function.

After the RBF is set up, the GA uses if for evaluating the objective functions of any design without requiring time-consuming CFD simulations.

Genetic Algorithm

The *multi-objective GA* incorporated into the optimization platform is based on the NSGA-II algorithm [15] and its structure is outlined in Figure 9. The GA starts by creating an initial population (popP) of Boxprop designs using the same parametrization as the overall optimization platform, and uses the meta-model to evaluate the values of the objective functions (C_T and η). The individuals of popP are then ranked into groups, and in this system Rank 1 contains the fittest individuals of the entire population, as is illustrated in Figure 10. Rank 1 contains individuals that are not dominated by any other individuals, and in this paper not being dominated means that no other individual has better thrust *and* efficiency. The ranking process goes through all points in the population, and if a given point has no other points inside its area of domination (see gray area in Figure 10), then that point belongs to Rank 1. After identifying all Rank 1 members, these individuals are removed from the ranking process. Rank 2 members are chosen from the remaining individuals using the same approach as Rank 1. The ranking procedure continues until all individuals have been ranked. In order to sort individuals inside each rank from best to worst, a *crowded distance measure* is used within each rank. This measure estimates how crowded the region is around each individual, and sorts all designs from best to worst based on how large the distance is to its neighbors. The crowded distance parameter ensures a good spread of solutions along each rank.

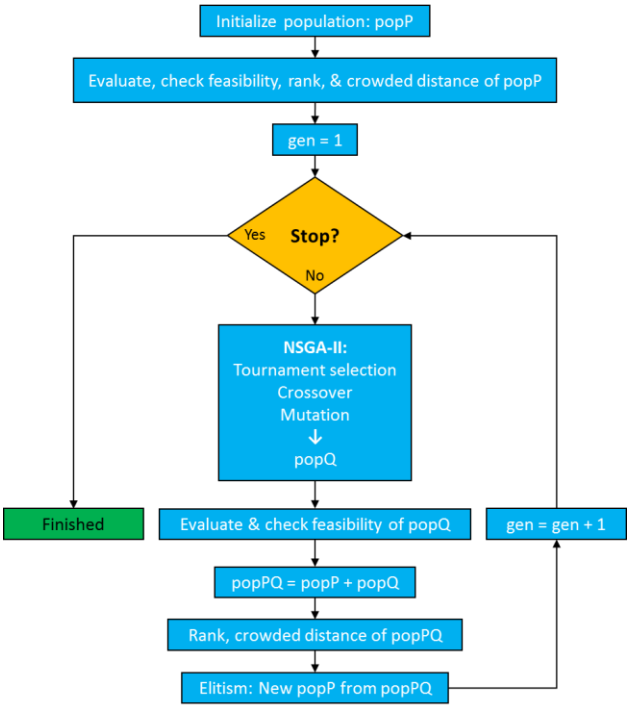


Figure 9 - Structure of the GA module. Image credit: [20]

The GA loop then starts with applying the processes of *selection*, *mutation*, and *crossover* on the population popP, thereby creating the new population popQ. Tournament selection with a size of two is used in this paper, and a one-point crossover process is used to represent biological reproduction. When the creation of popQ is finished, then its individuals are evaluated using the meta-model.

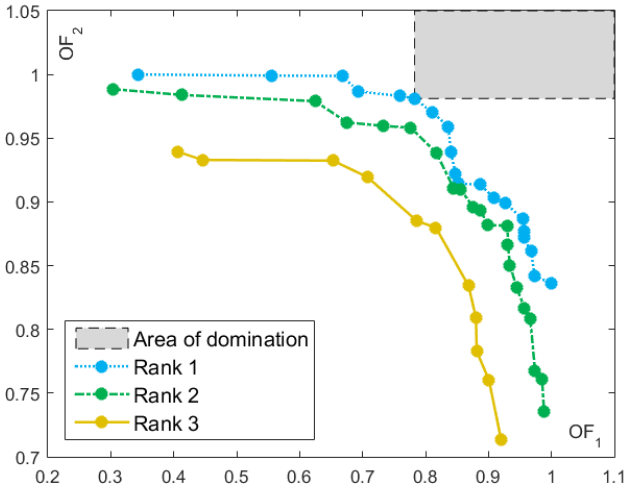


Figure 10 - Illustration of the different ranks produced by the GA, given generic objective functions (OF).

Elitism for single-objective optimization commonly saves the best individual in each generation, thus guaranteeing that the global optimum can never deteriorate. For multi-objective optimization there is no global optimum, rather all Rank 1 individuals are considered equally good. The optimization platform has therefore adopted the elitism approach used in the original NSGA-II algorithm, which is shown in Figure 11. Population PQ is first created by combining popP and popQ. Its individuals are then ordered from best to worst, first by rank, then by the crowded distance parameter within each rank. A new population popP is then created from this ordered set of designs. The GA loop continues for a fixed number of iterations or until a user-specified criteria has been met.

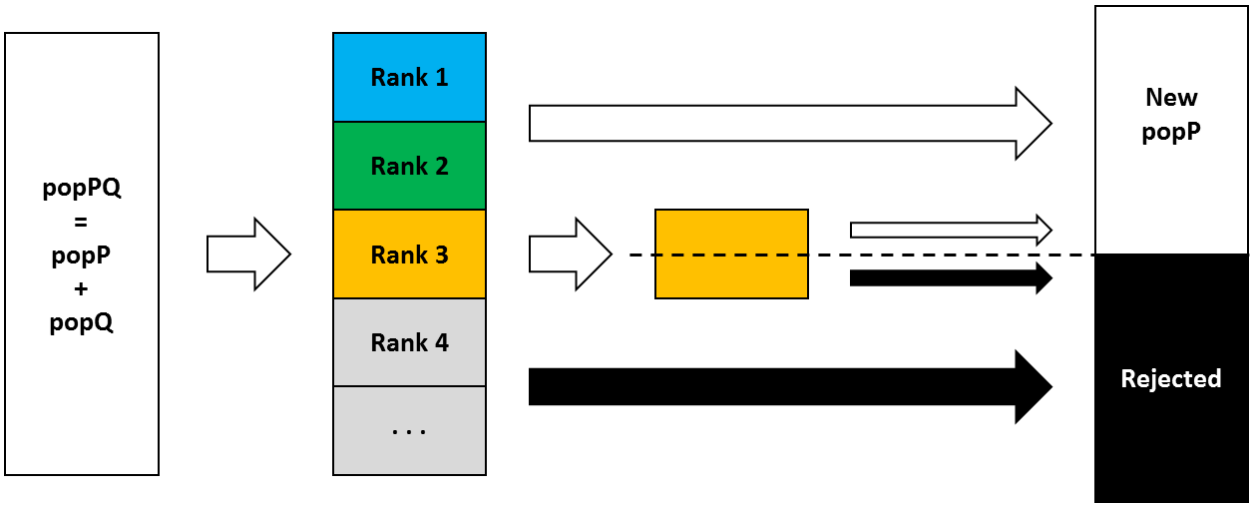


Figure 11 - The elitism approach used in the optimization platform. Population popPQ is formed from popP and popQ and its members are ordered using rank and the crowed distance parameter. A new version of popP is created by choosing individuals first by rank, then by the crowded distance parameter. Image credit: [20]

RESULTS

Table 2 - Propeller specification and operating conditions.

$D\ [m]$	HTR	J	$n\ [1/s]$	M_∞	$H\ [m]$	$p\ [Pa]$	$T\ [K]$
4.2672	0.4	3.56	14.64	0.75	10 668	23 922	218.9

The optimization platform described in the previous sections was unleashed on the Boxprop with the objectives of optimizing thrust and efficiency given the propeller specification and operating conditions given in Table 2. The diameter and hub to tip ratio (HTR) of the Boxprop correspond to what is found on modern CROR blades, and the rotational velocity at the tip is chosen to result in a relative Mach number of one. The advance ratio, Mach number and operating conditions of the blade are identical to the GPX701 (Figure 2) propeller found in a 2016 paper published by the authors [8], and will allow a comparison between it and the designs obtained from the optimization. The previously published GPX701 results were for a small scale version, but the propeller has been re-run at full scale, allowing a more suitable comparison with the results of the optimization.

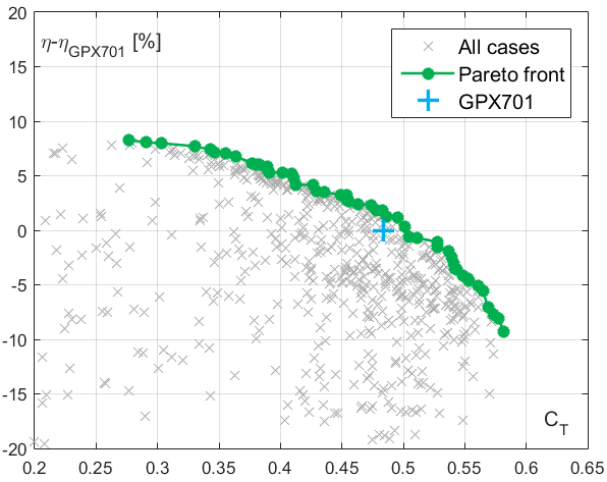


Figure 12 - All 740 propeller designs that have been evaluated using CFD, the obtained Pareto front, and the GPX701 performance. All efficiency values are shown relative to the efficiency of the GPX701 in order to illustrate the improvement.

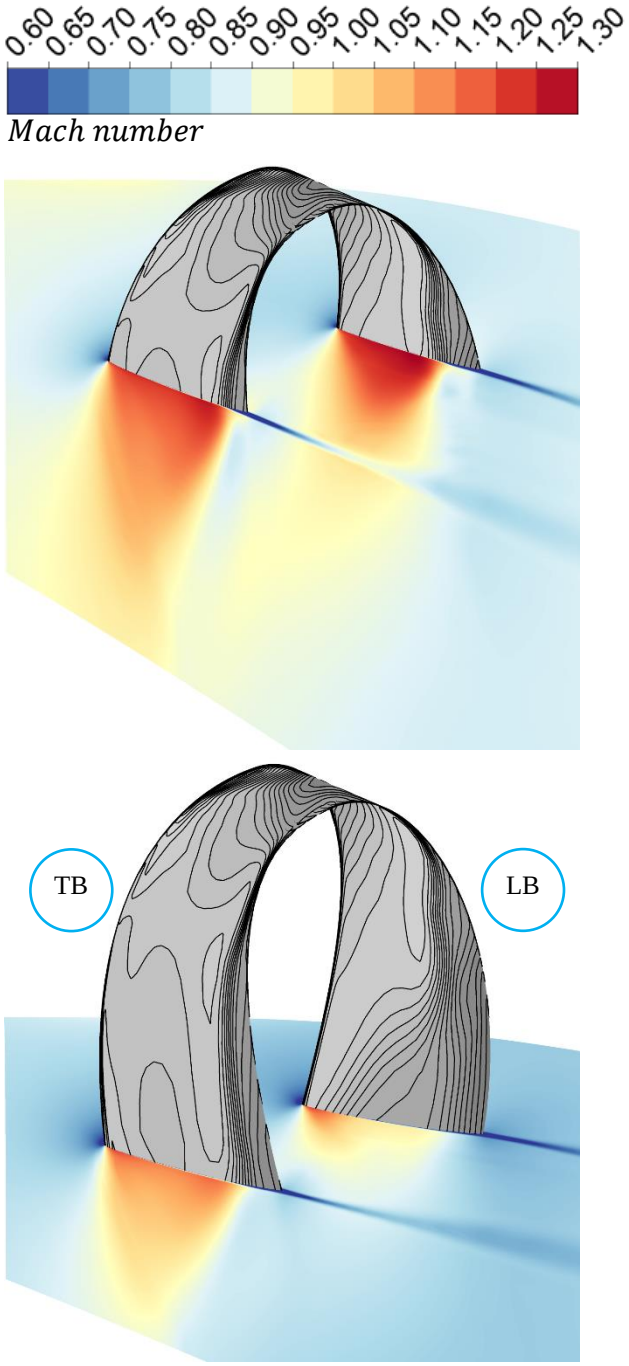


Figure 13 –Upper image: Mach number distribution and blade surface pressure isobars at $r/R = 0.5$ (lower image) and at $r/R = 0.75$ (upper image). Flow direction from left to right.

The overall results of the reduced optimization of the Boxprop are shown in Figure 12, which shows the 740 evaluated CFD cases for the optimization, the obtained Pareto front, and the performance of the GPX701 full scale propeller. At the thrust level of the GPX701, a modest improvement of approximately 1.3% in efficiency has been observed, despite the reduced set of optimization variables.

Flow features such as *shocks* are well represented (Figure 13), and the optimization has resulted in a placement of the shocks near the trailing edges of the blades, therefore preserving the low pressure region throughout the suction side and avoiding shock-induced separation, which would otherwise lead to increased losses. Additionally, the strength of the shocks is relatively low. The *blade wakes* are dissipated relatively quickly by the comparably coarse mesh, but this has been shown in previous work [8] to have little impact on the performance values. The GPX701 propeller from previous research produced more *swirl* as fraction of engine power (16.3%) in comparison to the optimized propeller shown in Figure 13, whose swirl is in the order of 12.7%. This difference in swirl partially explains the increase in efficiency that has been observed.

The leading blade (LB) and trailing blade (TB) of the designs along the Pareto front exhibit *lower blade interference* than the previously published GPX701, by virtue of the new Boxprop parametrization which allows the LB and TB to be moved farther away from each other. This has also been seen in sectional thrust profiles (not shown), which show that peak thrust has been shifted further out radially, which decreases swirl and results in a higher efficiency than for the GPX701.

DISCUSSION

The choice of *geometric parametrization* for the Boxprop has proven itself to be successful with respect to the blade parametrization objectives outlined earlier in this paper. These objectives encompassed the ability of the LB and TB to reach the optimal sectional loading, and the ability to decrease blade interference. The optimal sectional loading (as optimized by the GA) was enabled by individual, parametrized angle-of-attack and camber distributions for the LB and TB, while the decreased blade interference was achieved by the implemented stacking line parametrization, which allows the variation of the axial and tangential spacing of the LB and TB.

STL files have been experienced to be simple and fast to produce while offering sufficient accuracy for representing the Boxprop blade geometry. The geometric resolution of the STL surfaces can easily be adjusted, both in blade chord and span directions, allowing the user to tailor the resolution for the required analysis, and for diminishing the effects of its inherently discontinuous slope and non-existent curvature. The effects of slope and curvature are also present in the computational mesh, therefore the STL surfaces needs to have a higher resolution than the mesh itself. The developed blade parametrization and geometric representation is extendable to conventional propeller blades as well.

Regarding *discretization*, it was found that the Boxprop blocking structure was significantly more complex than for conventional propellers, and also required more support geometry to ensure high mesh quality. In terms of required *computational time* for convergence, it has been experienced that the Boxprop simulations converge more quickly than simulations of conventional propeller blades. A final explanation for this has not yet been obtained, but one hypothesis is that the time step calculated by the solver is larger for the Boxprop simulations, allowing convergence to be

reached faster. For the cases run in this paper, the total simulation time for each case ranged between approximately 60 to 90 minutes.

As was illustrated previously in this paper, the choice of basis and ϵ and for the *Radial Basis Functions* (RBF) yields very different interpolated values outside of the known data points. Most of the data points are used to create the RBF, but a fraction of the data points are used for error-estimation. In this way, a combination of basis and ϵ can be chosen that minimizes the interpolation error. Other types of meta-models could be implemented in the future, particularly if the meta-models include a built-in error estimation, as is found in *Kriging*.

The implemented *Genetic Algorithm* (GA) has shown itself very capable of finding designs along the Pareto front, and has been successfully tested for compressor optimization [20] and is currently being used in the design of a contraction for the wind tunnel described in [21].

An efficiency improvement of 1.3% has been achieved for the presented, reduced optimization for the same operating point as the previously presented GPX701 [8]. This increase in efficiency has been accomplished by decreasing the blade interference in the blade regions close to the tip, thereby allowing more thrust to be produced at higher radii (where it is more efficiently produced), and consequently generating less swirl (12.7% vs 16.3%). All results along the Pareto front have less blade interference than the GPX701, which is due to the blade halves having an increased axial spacing than previous Boxprop designs.

FUTURE SWEDISH-BRAZILIAN COLLABORATIVE USE

During a generic turbomachinery design process, general rules are applied by the designer to specify the blade angles, chord, height, tip desensitization, camber angle, thickness-to-chord ratio, pitch-to-chord ratio, and airfoil category. After 3D flow calculations, some undesirable flow structure can be observed in a specific region, generally close to the blade leading and trailing edges. This flow characteristic should be improved during the detailed design phase to reach better machine efficiency, thrust, and performance. Small geometric changes in key regions, radically influences the flow behavior. These geometric changes are very sensitive and require considerable time and attention from the designer. Optimization techniques brings a good tool to help and refine the turbomachine design, reducing the total design phase time and cost.

Each engineering problem, device, machine, and application have different constraints and physical aspects to take into account. In this work, a useful methodology was developed and tested, considering aero engines applications for industry and academia. To efficiently handle such problem formulations, optimization frameworks such as the one described in this paper become essential to provide interesting new solutions at reduced effort. The described optimization platform is clearly quite flexible and can be extended to operate on a number of applications in the area of turbomachinery. Collaboration that has a good foundation at both research organizations are:

- Fans including high speed military fans.
- Compressors
- Axial and radial turbines
- Pumps
- Hydraulic turbines such as the Francis and Kaplan turbines.

CONCLUSIONS

The results of this paper demonstrate the capability of the presented optimization platform, and its reduced application on the Boxprop has shown potential design improvements and insights which will definitely be incorporated into future Boxprop designs. The obtained designs show less blade interference, more efficiently loaded blades, and less produced swirl.

The methodology for geometry generation, meshing and optimizing is fast, robust, and readily extendable to other types of optimization problems, and paves the way for future collaborative research in the area of turbomachinery.

ACKNOWLEDGEMENTS

This work was supported by Sweden´s Aeronautical research programme sponsored jointly by the Swedish Armed Forces, the Swedish Defence Materiel Administration and the Swedish Governmental Agency for Innovation Systems. The authors would like to extend their gratitude towards Gonzalo Montero Villar for his exemplary work [20], first in collaboration with ITA, and then during the execution of his Master's Thesis in Chalmers University of Technology.

NOMENCLATURE

A_{ij}	*2nd order polynomial coefficient*	p	*Pressure [Pa]*
B_i	*2nd order polynomial coefficient*	r	*Propeller radial position [m]*
C_i	*2nd order polynomial coefficient*	$\boldsymbol{x}$	*RBF design variable vector*
C_T	*Coefficient of thrust*	$\boldsymbol{x}_j$	*Known design variable vector for the RBF*
D	*Propeller diameter [m]*	y_j	*Known objective function value*
H	*Altitude [m]*	$\hat{y}$	*Interpolated value from RBF*
J	*Advance ratio*	y^+_{ave}	*Average first node height*
M_∞	*Freestream Mach number*	ϵ	*RBF parameter*
N	*Number of LHS sample points/designs*	η	*Propeller efficiency*
OF	*Objective function*	κ	*Angle for control points [°]*
P_i	*Stacking line control points*	λ	*3D domain axial sizing parameter*
R	*Propeller tip radius*	μ	*3D domain radial sizing parameter*
T	*Atmospheric temperature*	ϕ	*RBF basis function*
d	*Distance for stacking line points [m]*	χ_i	*Design variable*
n	*Revolutions per second [1/s]*		

REFERENCES

[1] Larsson, L., “Analysis of concepts to reduce the environmental impact of aviation: high propulsive efficiency engines and contrail avoidance”, Chalmers University of Technology, 2014.

[2] Negulescu, C., "Airbus AI-PX7 CROR Design Features and Aerodynamics," SAE Int. J. Aerosp. 6(2):2013.

[3] Schnell RR, Yin JJ, Voss CC, Nicke EE. “Assessment and Optimization of the Aerodynamic and Acoustic Characteristics of a Counter Rotating Open Rotor”. ASME. J. Turbomach. 2012;134(6):061016-061016-15. doi:10.1115/1.4006285.

[4] Van Zante, D.E., 2015, “Progress in open rotor research: A U.S. perspective”. Proceedings of ASME Turbo Expo 2015. Paper number GT2015-42203.

[5] http://www.thinkdefence.co.uk/2012/05/interesting-engines-in-the-air/

[6] Axelsson, A., Göransson, A., Hung, C., Klipic, S., Landahl, J., Olofsson, J., Thor, D., 2012. ”Development of a Box-Bladed Propeller - Mechanical Analysis and Feasibility Study of the Concept.”, Tech. rep., Chalmers University of Technology, Department of Product Development. Part of the project course MPP126.

[7] Avellán, R. and Lundbladh, A., 2011, “Air Propeller Arrangement and Aircraft”, International Patent Application WO2011/081577A1, filed on Dec 28, 2009.

[8] Capitao Patrao, A., Avellan, R., Lundbladh. A., Grönstedt, T., 2016. ” Wake and loss analysis for a double bladed swept propeller”. Proceedings of ASME Turbo Expo 2016. Paper number GT2016-56540.

[9] L. Ellbrant, L-E Eriksson and H. Mårtensson 2012, Design of Compressor Blades Considering Efficiency and Stability Using CFD Based Optimization, Proceedings of ASME Turbo Expo, GT2012-69272, June 11-15, Copenhagen, Denmark.

[10] L. Ellbrant, L-E Eriksson and H. Mårtensson 2012, CFD Optimization of a Transonic Compressor Using Multiobjective GA and Metamodels, 28th ICAS Conference, September 23-28, Brisbane, Australia.

[11] M. Lejon, N. Andersson, L. Ellbrant, H. Mårtensson, T. Grönstedt, 2016, Optimization of Robust Transonic Compressor Blades, Proceedings of ASME Turbo Expo, GT2016-57236, June 13-17, Seoul, South Korea.

[12] Silva, O. F. R., Tomita, J. T., Bringhenti, C., Cavalca, D. F., 2014, “Hybrid Optimization Algorithm Applied on Multistage Axial Compressor Performance Calculations With Variable Geometry”, Engineering Optimization IV, Lisbon, Portugal, ISBN 978-1-138-02725-1.

[13] Silva, O. F. R., Tomita, J. T., Bringhenti, C., Cavalca, D. F., 2016, “Optimization Process Applied in the Preliminary Axial Flow Turbine Design”, Engineering Optimization V, Foz do Iguaçu, Brazil.

[14] Silva, O. F. R., Tomita, J. T., Bringhenti, C., Cavalca, D. F., 2016, “Evolutionary Algorithm and Sensitity Analysis Comparison Applied to Axial Flow Turbine”, Engineering Optimization V, Foz do Iguaçu, Brazil.

[15] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002.

[16] Avellán, R., Capitao Patrao, A., Lundbladh, A., and Grönstedt, T., 2015, “Preparing for a Proof-of-Concept of a Novel Propeller for Open Rotor Engines”, XXII International Symposium on Air Breathing Engines, Phoenix, AZ, United States, Paper No. ISABE 2015-22083.

[17] Violette, J.A., Sullivan, W.E., and Turnberg, J.E., 1984, “Large-Scale Advanced Prop-Fan (LAP) blade design”, Hamilton Standard, Windsor Locks, CT, United States, Paper No. NASA-CR-174790.

[18] Tomita, J. T., Silva, L. M., Silva, D. T., 2012, “Comparison Between Unstructured and Structured Meshes With Different Turbulence Models for a High Pressure Turbine Application”, Proceeding of ASME Turbo Expo 2012, GT2012-69990, Copenhagen, Denmark.

[19] Martin D Buhmann. Radial basis functions: theory and implementations. Cambridge monographs on applied and computational mathematics, 12:147–165, 2004.

[20] Montero Villar, G., 2016. "Aerodynamic Optimization of High Speed Propellers", Master Thesis, Chalmers University of Technology, 2016.

[21] B. Rojo, D. Kristmundsson, V. Chernoray, C. Arroyo, and J. Larsson, “Facility for investigating the flow in a low pressure turbine exit structure”, in 11th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, ETC 2015, 2015.P