TURBOMACHINERY BLADE DESIGN USING A NAVIER-STOKES SOLVER AND ARTIFICIAL NEURAL NETWORK

Stéphane Pierret
R.A. Van den Braembussche
Turbomachinery Department,
von Karman Institute for Fluid Dynamics,
Rhode-Saint-Genèse, Belgium

ABSTRACT
This paper describes a knowledge-based method for the automatic design of more efficient turbine blades. An Artificial Neural Network (ANN) is used to construct an approximate model (response surface) using a database containing Navier Stokes solutions for all previous designs. This approximate model is used for the optimization, by means of Simulated Annealing (SA), of the blade geometry which is then analyzed by a Navier-Stokes solver.

This procedure results in a considerable speed-up of the design process by reducing both the interventions of the operator and the computational effort. It is also shown how such a method allows the design of more efficient blades while satisfying both the aerodynamic and mechanical constraints. The method has been applied to different types of 2D turbine blades of which three examples are presented in this paper.

INTRODUCTION
The main goal when designing turbines or compressors is to achieve light, compact and highly efficient systems while reducing the cost and the duration of the design cycle in order to allow a rapid adaptation of the machines to the changing demands of the market.

The need to improve further the machine performance requires the use of 3D Navier-Stokes solvers during the design process. These solvers, however, do not indicate what geometry modifications are required to improve the blade performance. The search for optimized blades must therefore be guided either by an experienced designer or by a numerical method. This often requires a large number of Navier-Stokes computations, to evaluate many different blade geometries, before reaching a good solution satisfying both the aerodynamic and the mechanical requirements. Although this procedure allows the design of very efficient blades it is expensive in terms of computational and/or operator time. In order to keep the cost and the duration of the design process within reasonable limits, the design process must often be stopped as soon as an acceptable solution has been found.

NOMENCLATURE
\[\alpha\] Angle between \(I_{min}\) and \(X\)
\[a, d\] Neural network calculated and imposed output
\[C_{ax}\] Blade axial chord
\[C_{p}, C_{v}\] Specific heat coefficient
\[E\] Output error
\[F\] Transfer function
\[I_{min}, I_{max}\] Minimum and maximum inertia of blade section
\[n\] Degree of a Bézier curve
\[N_{B}\] Number of blades per row
\[P\] Static pressure
\[Re\] Reynolds number \(= \frac{\rho C_{ax} \sqrt{Re_{i}}}{\mu_{i}}\)
\[t\] Pitch
\[W\] Connection weight
\[X, Y\] Abscissa and ordinate
\[\beta\] Relative flow angle

\[\xi\] Loss coefficient
\[\gamma\] Learning rate
\[\kappa\] Momentum coefficient
\[N\] Neural network relation

\(\beta_{blade}, \lambda, \epsilon, \beta_{2}, \Delta \alpha, \Delta e_{i}, R_{i}, R_{e}, L_{1}, L_{2}, L_{4}, \alpha_{e}, \alpha_{p}\) Blade parameters (see figures 3 and 4)

Subscripts
0 stagnation conditions
1 inlet of computational domain
2 outlet of computational domain

Presented at the International Gas Turbine & Aeroengine Congress & Exhibition

This paper has been accepted for publication in the Transactions of the ASME
Discussion of it will be accepted at ASME Headquarters until September 30, 1998
without guarantee that it is also the optimum.

It is therefore important to reduce the number of Navier-Stokes computations needed to reach the optimum design. One way to speed up the design procedure is to use experience gained during previous designs for subsequent ones. However, the experienced designer may not always be available and/or may only have experience in a small number of turbine blade types.

The main objectives when developing the present method were: to guarantee optimum performance, to minimize the design time as well as the number of interventions of the designer and to have an evolutionary method able to use knowledge acquired during previous designs.

This paper describes the design method for 2D blade sections.

DESCRIPTION OF THE 2D DESIGN METHOD

The basic idea of the present method, of which a flow chart is shown in Figure 1, is to accelerate the design of new blades using the knowledge acquired during previous designs of similar blades. The core of this knowledge-based design system is therefore a DATABASE containing the input and output of all previous Navier-Stokes solutions i.e. the blade geometry (G), the flow field boundary conditions (BC) and the blade performance (\( \bar{P} \)) characterized by the efficiency, the turning and the Mach number distribution on the suction and pressure side.

Before starting the design of a new blade one has to specify the REQUIRED PERFORMANCES (aerodynamic and mechanical) namely: inlet and outlet flow angles \( (\beta_1, \beta_2) \), the pressure ratio \( (P_2/P_1) \), the Reynolds number to axial chord ratio \( (Re/Cax) \), the blade cross-section area \( (Area) \), the trailing edge radius \( (R_{te}) \), the moments of inertia \( (I_{min} \) and \( I_{max} \)) and the angle between \( I_{max} \) and the axial direction \( (\alpha) \) etc.

The first step of the design consists of proposing a new optimized GEOMETRY that will have to be analyzed by the Navier-Stokes solver. One starts by scanning the DATABASE to SELECT the sample that has a performance that is the closest to the required one. This geometry is then adapted, by means of an OPTIMIZATION procedure, to better satisfy the required performance. The last one uses a heuristic search procedure, called "simulated annealing", and an approximate model for the performance evaluation of the modified blade geometries. The approximate model used for this purpose is derived from the information contained in the DATABASE by means of an Artificial Neural Network (ANN). The latter is an interpolator which builds the response surface by LEARNING the relation between performance, boundary conditions and geometry:

\[
\bar{P} = \mathcal{N}(BC, G) \tag{1}
\]

After mapping the database samples, the ANN is able to generalize, meaning that it can PREDICT the performance of a new geometry, not present in the database.

During the second step, the new geometry, provided by the OPTIMIZATION, is evaluated by means of a NAVIER-STOKES SOLVER.

After the Navier-Stokes calculation, the geometrical parameters, the performance and the boundary conditions of this new sample are added to the DATABASE.

Finally, if the target performance has not been reached, a new iteration is started. Each new blade definition however has to be preceded by the LEARNING of the ANN using the new DATABASE. This one now contains a new geometry, closer to the requirements which allows an improvement of the relationship \( \mathcal{N} \).

As the time for ANN LEARNING is proportional to the number of training samples, it is sometimes of interest to build a sub-database (training database) containing only blade samples which are similar to the blade being designed. This "training database" is then used for the LEARNING of the ANN.

The main components of this procedure will be discussed in more detail in the next sections.

THE 2D NAVIER-STOKES SOLVER (TRA2D)

The Navier-Stokes solver used to predict the performance is the TRAF2D/3D solver developed by Arnone and al. (1994). The Reynolds-averaged Navier-Stokes equations are solved using finite volumes and a Runge-Kutta time integration scheme in conjunction with accelerating techniques such as local time stepping, residual smoothing and Full-Approximation-Storage (FAS) multigrid. Two-dimensional computations are usually performed with 12,000 points requiring a memory of 3 MB and 6 minutes of computational time on an Alpha workstation (type 500/333).
GEOMETRY MODEL

The blade is defined by a series of Bézier curves (Farin, 1993) connected into a closed contour in a way to ensure continuity at junction points.

A Bézier curve is specified by the coordinates of a series of points in space of which only the first and last lie on the curve they define. These points are known as the polygon points (or control points) of the curve, and the polygon constructed by connecting these polygon points with straight lines is known as the Bézier polygon of the curve. Bézier curves have three important properties: the degree of a Bézier curve, \( n \), equals the number of polygon points - 1, the tangents at the two end points are defined by the first and the last polygon lines and the curvature radius at end points depends only on the position of the first three polygon points.

Bézier curves have been chosen because the polygon formed by the polygon points mimics the Bézier curve and therefore allow an easy control of the curve. Although inflexion points can occur with Bézier curves, they are far less frequent than with polynomials or other analytical functions.

One has preferred to use a series of Bézier curves (composite curves) instead of a single one because composite curves allow imposing the location, the tangent and the curvature at key points on the blade surface, as there are: trailing edge location and the point on the suction side defining the throat. However, a special treatment at junction points is required to ensure the continuity, up to at least the second derivative. Although strict continuity of the third derivative is not necessary, the geometry model should, nevertheless, prevent large discontinuities in the slope of the curvature radius which are often responsible of velocity peaks on the blade surface. The use of composite curves facilitates the geometrical interpretation of the parameters defining the curve, allowing a better control of the geometry in order to avoid the generation of unrealistic blades.

This type of representation has been preferred to a point by point definition for three reasons: it ensures a very good continuity of the blade surface, the number of variables in the relation (1) is limited and the parameters defining the geometry have a physical meaning which facilitates imposing limitations on their variations and therefore allows restricting the design space.

**Parametric Blade Definition**

Several models have been evaluated. The one presented hereafter turns out to be the one whose parameters have the best geometrical interpretation avoiding the generation of unrealistic blades while having enough geometrical flexibility to represent a large number of blade types. Moreover the velocity distributions obtained with this model are very smooth.

The blade geometry is specified by four key points linked by four curves. The four key points are fully defined by the following parameters: \( R_{te}, \beta_2, \Delta \alpha, \delta_{te}, t, \varepsilon, Cax, \lambda \) (see Figure 2). The key points 1 and 2 are linked by a Bézier curve defined by 3 additional polygon points whose location is function of: \( \beta_{blade}, R_{te}, L_4, \alpha_{ss}, L_1 \) and the tangent at point 2 (see Figure 3). The key points 2 and 3 are linked using a Bézier curve defined by 2 additional polygon points. The last ones are located in such a way to ensure the continuity up to the third derivative at point 2 and continuity up to the first derivative at point 3. A Bézier curve with 3 additional polygon points is used to define the pressure side in the same way as the first part of the suction side. This curve is fully defined by the parameters: \( \beta_{blade}, R_{te}, L_4, \alpha_{ps}, L_3 \) and the tangent at point 2. Using \( R_{te} \) and \( L_4 \) for both the suction and the pressure side guarantees the continuity of the curvature radius at the leading edge. The trailing edge is defined by a part of a circle whose radius \( R_{te} \) is specified. The 2D blade geometry is thus fully defined by means of 15 parameters represented by \( G(n) : n=1,15 \).
Examples

Figure 4 shows 4 types of turbine blades generated with this geometry model. For each blade one parameter is changed and its influence on the blade shape is shown. This figure demonstrates that the method is capable of representing the various types of turbine blades encountered in industrial designs.

ARTIFICIAL NEURAL NETWORK

ANN are non-linear models that can be trained to map functions with multiple inputs/outputs (Cichocki and Unbehauen (1994)). Although the initial goal of artificial neural network is to imitate some brain functions, they can also be thought of as a very powerful interpolator.

The goal when using ANN is to construct an approximation of (1) using the information contained in the DATABASE. An ANN (Figure 5) is used here to relate performance ($\eta$, $\beta_2$) and the Mach number distribution given in 20 points on both blade sides ($M(m)$ : $m=1,40$) to the geometrical parameters and the aerodynamic boundary conditions. An ANN is composed of several elementary processing units called neurons or nodes. These nodes are organized in layers and joined with connections (synapses) of different intensity, called the connection weight ($W$) to form a parallel architecture. Each node performs two operations: the first one is the summation of all the incoming signals and the second one is the transformation of the signal by using a transfer function, very often defined by a sigmoidal function: $F(x) = \frac{1}{1+e^{-x}}$. This function introduces power series (given implicitly in the form of an exponential term) which avoids to make any hypotheses concerning the type of relationship between the input and the output variables. A network is generally composed of several layers; an input layer, zero, one or several hidden layers and one output layer.

Trained ANN are able to generalize which means that they give reasonable answers for input vectors that they have never seen. It is therefore possible to train an ANN on a representative set of input/output vectors without training the system on all possible cases.

Training by Back-Propagation of Errors

The LEARNING process consists of adjusting the connection weights and the bias in order to make the calculated output vector ($\vec{a}$) coincide with the imposed output vector ($\vec{d}$).

When a vector is presented at the network input, the signal is propagated to the output layer. Generally, the output vector provided by the network does not correspond to the imposed output vector associated with the input vector.

The error ($E$) is defined by the summation of the square of the errors on each output node:

$$E = \frac{1}{2} \sum_{i=1}^{m+2} (d_i - a_i)^2$$

This error is minimized by back-propagating the error to the network input. The back-propagation algorithm consists of finding better weight factors using the error ($E$) by adjusting each weight of the network proportionally and in the direction opposite to the error gradient with respect to this weight:

$$\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}}$$

Where $\gamma$ is a constant called the "learning rate", $i$ is the node number and $j$ is the layer number.

To perform the learning process, a set of input/output vectors (training samples) must be available (DATABASE) that the network will learn to predict. All the training samples are presented sequentially until convergence of the error.

Speeding-Up the Back-Propagation Learning Algorithm

The standard back-propagation training is characterized by a slow convergence. Numerous improvements of the stan-
dard back-propagation algorithm have been proposed in the
literature and two of them are used here.

The first consists of modifying the weights, not only propor-
tionally to the gradient of the error, but also proportion-
ally to the change of weight of the previous iteration (t-1) :
\[ \Delta w_{ij}^{(t)} = \gamma \Delta w_{ij}^{(t-1)} \]
where the momentum coefficient \( \gamma \) takes a value in the range \([0, 1]\). The purpose of this is to give the weight update a memory of its last update, providing a smoothing of the forces affecting the weight changes.

The second technique consists of adjusting the learning
coefficient \( \gamma \), during the process of convergence, to better
fit the local shape of the error function. The basic philo-
sophy of learning rate adaptation is the following: if the
gradient component has the same sign in two consecutive
steps \((t-1, t)\), the corresponding rate is increased and, if
the gradient component alternates, the learning rate is de-
creased. One of the simplest and most efficient algorithm
with locally adaptive learning rates has been proposed by
Silva and Almeida (reported by Cichocki and Unbehauen
1994). The algorithm employs batch back-propagation with momentum \( \kappa \) :
\[ \Delta w_{ij}^{(t)} = \gamma \Delta w_{ij}^{(t-1)} + \kappa \Delta v_{ij}^{(t-1)} \]

**OBJECTIVE FUNCTION**
A measure of the global performance of a blade geometry,
with respect to the requirements, is needed at several
steps in the design procedure: for the blade selection from
the database, during the optimization process and for the
convergence check after the Navier-Stokes calculation.

Efficiency is only one of the many considerations among
the design objectives. In addition the mechanical con-
straints and the imposed outlet flow angle must also be sat-
ished. The general approach to this problem is to build an
objective function (OF) which is the summation of penalty
terms in order to limit the violations of the constraints (Van-
derplaats, 1984).

The introduction of an additional penalty term, based on
the Mach number distribution, is justified by the 3 following
arguments:
- The losses may have multiple local minima (several blades
may have nearly the same losses) as well as large gradients
(mainly occurring when the boundary layer transition sud-
denly jumps from near leading edge to near trailing edge).
- Blades are also supposed to perform well at off-design con-
ditions. It is therefore important to introduce in the design
some constraints on the Mach number distribution which
ensure good performance of the blade over a wide range of
operating conditions.
- Current Navier-Stokes solvers have some uncertainty in
predicting the loss coefficient (mainly due to the uncertain-
ties in turbulence and transition modeling).

The penalty on the velocity distribution is derived from
the Mach number distribution in a limited number of points
on the blade surface \( \hat{M}(m, m=1,40) \). The criteria on the
Mach number distribution are then applied on the distri-
bution, reconstructed using a cubic spline passing through
these points. High penalty is given to velocity distributions
which are known of not being optimal.

There are 4 penalty terms on the Mach number distribu-
tion:
- Penalty if a minimum slope of the Mach number distribu-
tion on the suction side between the stagnation point and
the throat point (point 2) is not reached.
- Penalty on the maximum Mach number on the suction
side.
- Penalty based on the Pohlhausen parameter along the
suction side in order to stay away from separation. (This
type of penalty is very important in design because of
the uncertainties in predicting separation with existing
turbulence models).
- Penalty based on the slope of the Mach number on the
pressure side.

Finally the objective function is written as:
\[ OF = PMeca + PRP + P_t + PMach = OF(\hat{P}, \hat{G}, \hat{BC}) \] (3)
where \( PMeca \) stands for the penalties on the mechanical con-
straints, \( PRP \) stands for the penalties on the required per-
f ormance, \( P_t \) stands for the penalties on the loss coefficient
and \( PMach \) stands for the penalties on the Mach number
distribution.

It is also possible to use this method for the design
of blades with an imposed Mach number distribution by
adding, to the global objective function, a penalty term pro-
portional to the difference between calculated and imposed
Mach number distribution. This penalty is the summation
of the square of the Mach number error along the blade
surface.

Hence, depending on the type of design problem, we can
minimize the losses or impose a Mach Number distribution
by switching on or off some penalty terms.

The development of the objective function is a delicate
and long procedure. It may contain a lot of terms quanti-
fying as well design objectives as manufacturing constraints
and cost.

**OPTIMIZATION**
The goal of the optimization is to find the geometry which
minimize the objective function while evaluating the blade
performance with the simplified model \( \hat{P} = N(BC, G) \) ob-
tained from ANN.

The choice of the optimization algorithm is mainly based
on two considerations:
- the existence of many local optima in the design space may
pose a problem for gradient type optimization methods.
- the efficiency of the optimization algorithm in terms of
number of function evaluations is of far less importance
when using a simplified prediction model than if a detailed
Navier-Stokes computation was needed at each step of the
optimization process.
For such optimization problems stochastic techniques offer a valid alternative to conventional gradient methods. Simulated Annealing (SA) algorithm, based on the analogy with the annealing of solids (van Laarhoven and Aarts, 1987), is such a method.

There is no real definition of what optimization based on simulated annealing really is. The only requirements are: sufficient randomness must be inherent in the point selection process so that a wide field can be covered; more optimal points must be somewhat favored over less optimal points; and the degree of randomness must slowly decrease.

RESULTS
This design procedure has been successfully tested on a large number of designs and will be illustrated here by 3 examples.

Design of a reaction-type blade
The design requirements of this turbine blade are summarized in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Imposed</th>
<th>After 4 modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet flow angle (β1)</td>
<td>0.0°</td>
<td>-</td>
</tr>
<tr>
<td>$P_2/P_{01}$</td>
<td>0.6629</td>
<td>-</td>
</tr>
<tr>
<td>$Re/Cax$ (1/m)</td>
<td>9.3 $10^7$</td>
<td>-</td>
</tr>
<tr>
<td>$k = Cp/Cv$</td>
<td>1.287</td>
<td>-</td>
</tr>
<tr>
<td>Cax (m)</td>
<td>0.07615</td>
<td>-</td>
</tr>
<tr>
<td>NB (number of blades)</td>
<td>60</td>
<td>-</td>
</tr>
<tr>
<td>Blade radial location (m)</td>
<td>0.7781</td>
<td>-</td>
</tr>
<tr>
<td>Trailing edge radius (m)</td>
<td>2.0 $10^{-4}$</td>
<td>-</td>
</tr>
<tr>
<td>Area (m$^2$)</td>
<td>1.73 $10^{-4}$</td>
<td>1.67 $10^{-4}$</td>
</tr>
<tr>
<td>$I_{min}$ (m$^4$)</td>
<td>9.20 $10^{-9}$</td>
<td>8.74 $10^{-9}$</td>
</tr>
<tr>
<td>$I_{max}$ (m$^4$)</td>
<td>1.07 $10^{-8}$</td>
<td>1.03 $10^{-8}$</td>
</tr>
<tr>
<td>alpha</td>
<td>45.0°</td>
<td>48°</td>
</tr>
<tr>
<td>Outlet flow angle (β2)</td>
<td>-73.4°</td>
<td>-73.2°</td>
</tr>
<tr>
<td>Loss coefficient (ζ)</td>
<td>0.0 %</td>
<td>1.54 %</td>
</tr>
</tbody>
</table>

Table 1: Mechanical and Aerodynamic requirements

The design starts with a database containing 20 samples and an ANN with two layers and 10 nodes in the hidden layer. The computational time needed to train the ANN is less than 10% of the time needed by the Navier-Stokes solver.

The initial blade selected in the database, whose Mach number distribution and geometry are presented in Figures 6 and 7, already has a good Mach number distribution. This is one advantage of the method which uses as first guess a blade shape which has already been optimized under similar operating conditions. However, the pitch is slightly larger than the required one, the pressure ratio is smaller than the imposed one and not all the mechanical constraints are respected.

This initial blade is then optimized by means of ANN and SA for the imposed requirements and analyzed by the Navier-Stokes solver. The result after the first modification and first Navier-Stokes solver is presented in Figures 6 and 7. The Mach number distribution is smooth but the maximum value at $X/Cax = 0.8$ is quite large. It introduces a too large diffusion on the rear part of the suction side.

Near optimum performance has been obtained already after four modifications (four Navier-Stokes computations) (see Figure 8). The main improvements on the Mach num-
Design of an impulse-type blade

The design of impulse blades is usually very difficult because of the small pitch to axial chord ratio, resulting in a strong interaction between the velocity distributions on the pressure and suction side and because of the small acceleration of the velocity from the leading edge to the trailing edge.

Fig. 9: Mach number distributions on the blade surface

The design requirements of this 2nd example are displayed in Table 2. This design started with a database containing 50 samples. The initial blade sample selected from the database has a slightly lower outlet isentropic Mach number (0.537 instead of 0.6 required) as well as a higher outlet flow angle than the required one (-70.1 deg. instead of -68.9 deg.).

A near-optimum solution is already obtained after 7 modifications without any intervention of the operator. The suction side Mach number distribution increases almost linearly from the leading edge until the throat. The maximum Mach number on the suction side is only slightly higher than the outlet Mach number thus limiting the diffusion on the rear part of the blade. (Figures 9 and 10). The aerodynamic and mechanical requirements are satisfied within the imposed tolerances (see Table 2).

Design of an impulse-type blade with an imposed Mach number distribution

As mentioned previously, the method also allows defining a blade with a prescribed Mach number distribution on the suction and pressure side. The penalty term on the losses and the penalty on the Mach number criteria are replaced by a penalty expressing the difference between the required and the real Mach number distribution. The penalties on the outlet flow angle and on the mechanical requirements are unchanged.

The design requirements are displayed in Table 3. To make sure that a solution exists, the Mach number distribution of an existing blade has been imposed as target distribution together with the corresponding outlet flow angle and mechanical constraints. Of course this existing blade is not stored in the database.

---

### Table 2: Mechanical and Aerodynamic requirements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Imposed</th>
<th>After 7 modif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet flow angle ((\beta_1))</td>
<td>55.0°</td>
<td>-</td>
</tr>
<tr>
<td>(P_2/P_0)</td>
<td>0.7978</td>
<td>-</td>
</tr>
<tr>
<td>(Re/C_T (1/m))</td>
<td>9.3 (10^7)</td>
<td>-</td>
</tr>
<tr>
<td>(k = C_p/C_v)</td>
<td>1.287</td>
<td>-</td>
</tr>
<tr>
<td>(C_a) (m)</td>
<td>0.03705</td>
<td>-</td>
</tr>
<tr>
<td>NB (number of blades)</td>
<td>192</td>
<td>-</td>
</tr>
<tr>
<td>Blade radial location (m)</td>
<td>0.708</td>
<td>-</td>
</tr>
<tr>
<td>Trailing edge radius (m)</td>
<td>3.0 (10^{-4})</td>
<td>-</td>
</tr>
<tr>
<td>Area ((m^2))</td>
<td>3.10 (10^{-4})</td>
<td>3.01 (10^{-4})</td>
</tr>
<tr>
<td>(I_{\min} (m^4))</td>
<td>6.20 (10^{-9})</td>
<td>6.30 (10^{-9})</td>
</tr>
<tr>
<td>(I_{\max} (m^4))</td>
<td>2.70 (10^{-8})</td>
<td>2.70 (10^{-8})</td>
</tr>
<tr>
<td>alpha</td>
<td>-16°</td>
<td>-</td>
</tr>
<tr>
<td>Outlet flow angle ((\beta_2))</td>
<td>-68.9°</td>
<td>-69.1°</td>
</tr>
<tr>
<td>Loss coefficient ((\xi))</td>
<td>0.0 %</td>
<td>3.86 %</td>
</tr>
</tbody>
</table>

---

### Table 3: Mechanical and Aerodynamic requirements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Imposed</th>
<th>After 8 modif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet flow angle ((\beta_1))</td>
<td>55.0°</td>
<td>-</td>
</tr>
<tr>
<td>(P_2/P_0)</td>
<td>0.8051</td>
<td>-</td>
</tr>
<tr>
<td>(Re/C_T (1/m))</td>
<td>7.0 (10^7)</td>
<td>-</td>
</tr>
<tr>
<td>(k = C_p/C_v)</td>
<td>1.287</td>
<td>-</td>
</tr>
<tr>
<td>(C_a) (m)</td>
<td>0.03705</td>
<td>-</td>
</tr>
<tr>
<td>NB (number of blades)</td>
<td>192</td>
<td>-</td>
</tr>
<tr>
<td>Blade radial location (m)</td>
<td>0.708</td>
<td>-</td>
</tr>
<tr>
<td>Trailing edge radius (m)</td>
<td>3.0 (10^{-4})</td>
<td>-</td>
</tr>
<tr>
<td>Area ((m^2))</td>
<td>3.10 (10^{-4})</td>
<td>3.07 (10^{-4})</td>
</tr>
<tr>
<td>(I_{\min} (m^4))</td>
<td>6.20 (10^{-9})</td>
<td>5.97 (10^{-9})</td>
</tr>
<tr>
<td>(I_{\max} (m^4))</td>
<td>2.70 (10^{-8})</td>
<td>2.64 (10^{-8})</td>
</tr>
<tr>
<td>alpha</td>
<td>-14°</td>
<td>-</td>
</tr>
<tr>
<td>Outlet flow angle ((\beta_2))</td>
<td>-68.9°</td>
<td>-68.7°</td>
</tr>
<tr>
<td>Loss coefficient ((\xi))</td>
<td>-</td>
<td>4.23 %</td>
</tr>
</tbody>
</table>
After 8 modifications an almost perfect agreement is obtained on both the suction and pressure side with only a discrepancy close to the leading edge on the pressure side. The main reason for this discrepancy is the fact that the target Mach number distribution is not imposed in a sufficient number of points in the region $X/C_{ax} < 0.1$.

CONCLUSIONS

We have shown that, using the method presented in this paper, only a few Navier-Stokes computations are needed to define an optimized blade satisfying the aerodynamic and the mechanical requirements.

This design method allows the exploration of more design options in a given period of time than with traditional methods.

The efficiency of the method results from:
- the use of a robust geometry model which avoids the generation of unrealistic blades but which nevertheless still has enough flexibility to reach a large number of blade types.
- the use of an artificial neural network and a database able to acquire experience from previous designs and efficiently use that experience for subsequent designs.
- A fully automated design procedure without operator intervention is possible mainly due to the development of an objective function which translates the judgment of a specialized designer (concerning the quality and the reliability of the solution), and therefore the market demand, into a single number which can be handled by a computer.

The optimization algorithm, based on simulated annealing, guarantees that the system will not be trapped in a local minimum.

The method can be used with any Navier-Stokes solver. Although this work has so far focused on two-dimensional turbine blades, the method can also be applied to the design of compressor blades.

This method can also be extended to the design of three-dimensional blades. The design can be realized in 3 steps:
- several 2D sections are designed and then stacked and analysed by a 3D Navier-Stokes solver.
- the changes due to 3D effects are then introduced as a correction to the 2D requirements and the 2D sections are redesigned.
- as soon as a database containing a sufficiently large number of 3D computations is available, 3D optimization is possible by extending the approximate model to the prediction of 3D performances.

ACKNOWLEDGMENTS

The first author is financially supported by the Belgian "Fonds pour la formation à la Recherche dans l'Industrie et l'Agriculture" (FRIA). The von Karman Institute acknowledges the computer support provided by Ansaldo Energia s.p.a. Thanks also to Dr. A. Arnone for making his solver available.

REFERENCES


