Design optimization of a multistage axial turbine using a response surface based strategy

Carlo Cravero¹, Paolo Macelloni²

¹ DIMSET – CFD Group, Università di Genova, Genova, Italy, cravero@unige.it
² DIMSET – CFD Group, Università di Genova, Genova, Italy, paolo.macelloni@unige.it

Abstract
An optimization design approach applied to a 4 stage axial flow turbine (AGARD E/TU-4 testcase) is presented. For this purpose the commercial software ModeFRONTIER has been adopted. A workflow based on geometrical parameterization of the system, a two dimensional throughflow flow solver and a multiobjective genetic algorithm has been built. Only blade sections restaggering has been considered leaving the meridional channel and the number of blades for each single row unchanged. This in order to re-design an existing turbine with the possibility for a direct substitution of the original blade with the new ones. The optimization goal is the maximization of the total-to-static turbine efficiency (at design operating conditions) keeping the massflow rate unchanged. Two optimization approaches have been considered: with the direct use of the flow solver or with metamodels based on a RSM. The multilayer perceptrons - Artificial Neural Networks - have been used as RSM. Since the number of input variables is high and the design space not easy to fill, a Design of Experiment sensitivity analysis has been carried out, in order to select the best combination of algorithms for an efficient design space exploration. A quasi random algorithm (Sobol) coupled with a statistical distribution algorithm (Normal distributed Latin Hypercube sampling) have been used as space fillers. The DoE has been validated with the flow solver in order to obtain the training dataset for the supervised learning perceptrons. Using the ANNs, the flow solver has been by-passed and a quick and accurate optimization has been carried out, based on metamodels for massflow, total-to-static efficiency and power output. The obtained optimum turbine has been compared to the original and validated using the throughflow solver. The proposed approach has demonstrated its potential for the optimization of multistage axial turbines and it is considered a very useful candidate for design optimization of multistage axial turbomachinery with a high number of stages.

Keywords: Artificial Neural Networks, DoE techniques, Turbomachinery, NSGA-II

1. Introduction
Industrial production rates require nowadays very low-time consuming design strategies, in order to improve flexibility and reduce design timescales. In many fields of industrial applications multiobjective design optimization techniques are increasingly used, including turbomachinery development, both for industrial design and redesign approaches. Modern 3D RANS flow solvers can be adopted for the turbine stage simulations with high accuracy, allowing detailed descriptions of the flow in the computational domain. Moreover current hardware resources allow the simulation of the entire multistage configuration to be performed thanks to the increased use of code parallelization. Nevertheless the unsteady, viscous, compressible and fully-3D nature of the flow inside the stages requires long computational runtimes (hours or days) even if enhanced HPC clusters are adopted. Nevertheless in such a scenario the optimization approach with a direct use of high-fidelity 3D flow solvers is still not practical for the design optimization of a multistage turbomachinery. This is due to the following main reasons: high number of independent variables, large number of individuals required by a Multiobjective Genetic Algorithm, long runtime for every single goal function evaluation. To overcome the above limits many soft-computing techniques are increasingly adopted with the use of metamodels. Algorithms like the Artificial Neural Networks (ANN) [1, 2] are introduced to model and simulate complex systems in combination with genetic algorithms (GA) [3] to explore the design space to find the optimum. RSM techniques, Artificial Neural Networks, and genetic algorithms are therefore adopted to develop optimization strategies that will substitute the direct use of the flow solver (i.e. 3D RANS) with the metamodel with a dramatic reduction of the timescales. With the above approach a critical issue is the DoE technique used to fill the design space and to feed the RSM model. This aspect is discussed in the paper and a sensitivity analysis with several DoE approaches is presented. Different applications of ANN based optimizations for turbomachinery design have been presented: efficiency maximization of 3D blade profile design with mechanical constraints [4], simulation of steam turbine cascades and stage design [5, 6, 7, 8], compressor applications [9] and aerodynamic design [10]. With respect to the above applications, the present work focuses on the entire multistage turbomachine that, in case of steam turbines can have more than 20 stages. As previously mentioned the direct use of a 3D solver is impractical in this case and
therefore a simplified flow solver is used for the performance analysis of the turbine. A 2D simulation using a throughflow (TF) code [11, 12, 13] has been used for the turbine analysis. The 4 stages axial turbine from the AGARD E/TU-4 testcase [14] has been considered as reference configuration to be optimized and to validate the TF code. A multiobjective genetic algorithm NSGA-II [15], previously used for axial compressor blade optimization [16, 17] has been embedded into the design strategy. Two design optimization approaches have been developed: with the direct use of the flow solver, with the use of an ANN-based surrogate model for the turbine performance. Although the flow-solver based optimization has been possible (low runtime simulations) the ANN - response surface model has demonstrated to be a faster approach reaching practically the same optimum in much lower runtimes. The proposed metamodel based optimization strategy is therefore considered a valid tool for multistage turbine redesign.

2. The test multistage axial-flow turbine and the simulation method

The 4-stage low-speed axial-flow turbine extensively documented in literature (AGARD E/TU-4 testcase) [14] has been considered as reference test-case to develop the proposed optimization approach. This test is probably the only multistage turbine available in the open literature having a complete set of both geometrical data and experimental results. The turbine has been tested at both design and off-design conditions. For the above reasons it is an ideal candidate for testing the simulation method used and the proposed optimization procedure for multistage turbines. The main design data and design operating conditions of the turbine [14] are:

- Coupling power: 703 kW
- Rotational speed: 7500 rpm
- Air flow rate: 7.8 kg/s
- Inlet total pressure: 260000 Pa
- Inlet total temperature: 413 K
- Outlet pressure: 102200 Pa
- Outlet temperature: 319 K
- Total-to-total efficiency: 91.3%
- Expansion ratio: 2.5:1
- Degree of reaction: 0.5
- Hub/tip ratio at outlet: 0.525

In figure 1 a view of the meridional channel with experimental control stations is shown.

![Figure 1: The meridional channel of the E/TU-4 turbine.](image)

To simulate the fluid-dynamics of the turbine, a throughflow code has been used. The throughflow computational approach is an industrial standard for multistage turbines and it has been successfully used for the optimized design of single axial-flow turbine stages [13]. It is a very interesting tool because of its very low computational demand (short runtimes of the order of seconds). The accuracy in predicting the turbine performance depends on the loss correlation set implemented. The TF code used is based on the Streamline Curvature Methodology (SLCM) proposed by Denton [11]. It implements classical loss correlations available in literature [18, 19, 20]. To understand the accuracy of the code a preliminary set of computations has been performed to compare the predicted performance to the available experimental data. In figure 2a – 2b the spanwise distributions of total pressure and temperature (respectively) at the turbine outlet obtained with the TF code are compared to the experimental data and a good agreement is shown. In table 1 the overall turbine performance at design conditions obtained with the TF code are reported and in table 2 several off-design conditions are considered at different outlet pressures and rotational speed with respect to the design values [14]. The above numerical values are plotted...
in figure 3, where the obtained results are compared to the reference experimental data.

Table 1: Global performances at design conditions obtained with the TF code computations.

<table>
<thead>
<tr>
<th>Power P</th>
<th>[kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>664.73</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Massflow rate m'</th>
<th>[kg/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.79</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total-to-total efficiency</th>
<th>[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.99</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total-to-static efficiency</th>
<th>[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>89.36</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-dimensional rotational speed n/n_d [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 2: Global performances at off-design conditions obtained with TF code computations.

<table>
<thead>
<tr>
<th>m'/m'_d</th>
<th>n/n_d</th>
<th>η_t</th>
<th>η_s</th>
<th>P [kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.81</td>
<td>1.00</td>
<td>0.8988</td>
<td>0.8817</td>
<td>407.77</td>
</tr>
<tr>
<td>0.57</td>
<td>1.00</td>
<td>0.8187</td>
<td>0.7776</td>
<td>154.93</td>
</tr>
<tr>
<td>0.51</td>
<td>1.00</td>
<td>0.7839</td>
<td>0.7317</td>
<td>111.80</td>
</tr>
<tr>
<td>0.69</td>
<td>0.75</td>
<td>0.8987</td>
<td>0.8862</td>
<td>278.71</td>
</tr>
<tr>
<td>0.49</td>
<td>0.75</td>
<td>0.8808</td>
<td>0.8575</td>
<td>120.97</td>
</tr>
<tr>
<td>0.40</td>
<td>0.75</td>
<td>0.8409</td>
<td>0.8030</td>
<td>71.02</td>
</tr>
</tbody>
</table>

Figure 2a – 2b: Comparison between experimental and calculated \( p_t \) and \( T_t \) distributions at station 4 (outlet).

Figure 3: Comparison between experimental and computed \( \eta_s \) at design and off-design conditions.

The results obtained with the TF code are the state-of-the-art for the throughflow simulation method [12, 21] and are considered fully acceptable for the scope of the present work. The TF solver has therefore been selected as reference tool for the performance prediction of multistage axial turbines in the optimization process.

3. The optimization problem

3.1 The system parameterization

The optimization procedure needs a set of independent variables to work with. This set is obtained from the parameterization of the system and it is a fundamental step for any design optimization approach. Too few variables and the optimization procedure will converge quickly, but the system will be only roughly described. A sophisticated and detailed parameterization will result in a high level of freedom for the optimization algorithm, but a much slower iterative process will be produced. Moreover the choice of the independent variables
A constrained optimization problem is set up for the maximization of the total-to-static turbine efficiency with given inlet total flow conditions (pressure and temperature) and outlet static pressure. The mass-flow rate is constrained to the design value with a tolerance of ±0.02 kg/s.

3.2 The optimization procedure

The E/TU-4 turbine case with four stages has a total number of 24 variables (3 parameters per blade row $\Delta \beta_H$, $\Delta \beta_M$, $\Delta \beta_T$). A restagger of ±5 degrees with respect to the original design is considered as design space for the variables. The above domain has been discretized with a 0.05 deg. resolution, maintaining coherence with actual industrial production standards in terms of machining tolerances and metrological measurement resolutions. The use of discrete steps in variable change needs an evolutionary algorithm able to accept non-continuous variables. The DoE technique was adopted for the starting generation of samples. The genetic algorithm, coupled with the model (or with the metamodel) starts from the initial population generated with the DoE and, after the evaluation of the system response, it sets the input variables to update the system in order to optimize the user defined objective functions.

In case of a Response Surface-based optimization, the design space exploration through the DoE is a crucial aspect and it will be discussed in the following parts.

During the optimization loop the metal angles of the original turbine are algebraically summed to the restaggering and it will be discussed in the following parts. The independent variables are the restaggering angles at hub, midspan and tip of each blade, and, considering the non-dimensional blade height $0<h<1$, the restaggering function is:

$$\Delta \beta = ah^2 + bh + c$$  \hspace{1cm} (1)

where $a$, $b$ and $c$ are the coefficients obtained from the restaggering angles at hub, mid and tip as defined in Eq. (2):

$$\begin{cases} a = 2(\Delta \beta_T + \Delta \beta_H - 2\Delta \beta_M) \\ b = 4\Delta \beta_M - \Delta \beta_T - 3\Delta \beta_H \\ c = \Delta \beta_H \end{cases}$$  \hspace{1cm} (2)

The total number of independent variables is therefore three times the number of blade rows and, in case of large multistage turbines (i.e. 20 stages) it results in a quite high number (120 variables). The present problem is therefore highly significant for turbomachinery applications and it is representative of a class of optimization problems with high number of variables. In order to check if the dimension of the problem could be reduced, the Student’s t-test [22] was applied to investigate the influence of all input variables on output; from the t-test emerged that none of the variables is negligible. Therefore the parameterization adopted is meaningful and not redundant. An alternative practice to check the parameterization is to adopt a variable selection technique [23, 24].

4. Flow solver based optimization

The very short runtime of the TF code allows its direct use in the optimization loop for the turbine performance
simulation. The computational effort of a more accurate 3D RANS solver would require a different order of magnitude and this severely limits its use for the present application. For the test turbine, the 24 input variables are grouped into 3 vectors (8 variables each) in the ModeFRONTIER platform. Each vector contains the restaggering angle value at hub, midspan and tip of the blade. The advantage to use vectors in the ModeFRONTIER workflow is its flexibility: increasing the number of stage for the turbine, the only change needed will be the vectors dimension increase; the program workflow structure will be not modified. There are 3 output variables: total-to-static efficiency ($\eta_{ts}$), massflow rate ($m'$), and power output ($P$). Only the $\eta_{ts}$ is maximized, while the massflow rate $m'$ is constrained. The output Power $P$ is simply monitored, being function of the other variables:

$$P = \eta_{ts} \cdot m'(h_{1} - h_{2ts})$$

The isoentropic enthalpy drop is fixed by the boundary conditions imposed to the flow solver.

The NSGA-II genetic algorithm was set to iterate for 1000 generations, with a crossover probability of 0.9. Each generation was made of 100 samples, for a total of 100000 throughflow calculations. The optimization run took 14 hours. Actually a good convergence was already reached after 10000 calculations (1 h and 30 minutes approx.), but constantly and slightly improving until 100000 calculations, where the optimum was reached.

The following performance change has been obtained with respect to the reference turbine:

- $\eta_{ts}$ increased by 0.31%
- $P$ increased by 0.40%
- $m'$ increased by 0.1% (satisfies the constraints)

The obtained results are in agreement with other applications using a similar parameterization but a different strategy [6].

5. Artificial Neural Network based optimization

A different optimization approach has been developed using a RSM instead of the direct use of the TF code for the turbine performance prediction. The Artificial Neural Networks (ANN) have been used as metamodels for the three output variables $\eta_{ts}$, $m'$, $P$. The ANN metamodels were used in order to:

- verify runtimes and optimization level achievable with metamodels;
- develop a general approach for virtual optimization to optimize a complete turbine with a very high-number of stages.

As discussed before, the response surface based optimization requires a more accurate and rich DoE set, that must be efficient in design space coverage; for this purpose low discrepancy sampling methods are typically adopted as design space fillers [27].

5.1 The DoE technique: sensitivity analysis and application

The generation of a well-distributed population of samples in the input variable domain (design space) and the validation of the samples through the model allow a high fidelity approximation of the model outputs. The better are distributed the samples in the design space, the lower number of validations are required to estimate the model responses.

A good space filling plays a keyrole in metamodels-based optimization for two reasons:

- in order to create a set of samples that "sustain" the metamodel;
- in order to have a faster convergence of the algorithm.

Two space filling techniques were selected for the DoE generation after a comparison of convergence benchmark between many DoE generation algorithms.

5.1.1 DoE benchmark: the selection of the algorithms

In order to adopt an efficient space filling technique, a benchmark between many different sampling methods was carried out. Some algorithms are more efficient than others when a high number of variables is considered; for this purpose were used both classical and modern DoE techniques [28].

<table>
<thead>
<tr>
<th>Feasibility Chart</th>
<th>Sampling Algorithm</th>
<th>Total n. of converged configurations on 2600</th>
<th>Converged feasible</th>
<th>Converged unfeasible (broken constraints)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Latin Hyp. (Normal distribution)</td>
<td>2198</td>
<td>632</td>
<td>1566</td>
</tr>
<tr>
<td>2</td>
<td>Sobol</td>
<td>2188</td>
<td>219</td>
<td>1969</td>
</tr>
<tr>
<td>3</td>
<td>Hammersley</td>
<td>2143</td>
<td>201</td>
<td>1942</td>
</tr>
<tr>
<td>4</td>
<td>Faure</td>
<td>2180</td>
<td>197</td>
<td>1983</td>
</tr>
<tr>
<td>5</td>
<td>Random</td>
<td>2189</td>
<td>192</td>
<td>1997</td>
</tr>
<tr>
<td>6</td>
<td>Latin Hyp. (Uniform distribution)</td>
<td>2141</td>
<td>176</td>
<td>1965</td>
</tr>
</tbody>
</table>
Six different DoEs of 2600 samples were generated using the following algorithms: Random, Faure [29], Hammersley [30], Sobol [31, 32, 33] and Latin Hypercube with uniform and with normal distributions [34]. Each sample represents a turbine configuration (identified by its distribution of stagger angles). Not all the 2600 samples converged during the validation phase, and not all the samples that reached convergence satisfied the massflow constraints (giving unfeasible designs). In table 3 is reported a feasibility chart of the benchmark results.

As clearly results, the Latin Hypercube approach with normal distribution has the largest number of feasible configurations due to the Gaussian and normalized-on-the-mean-value distribution of the input variables. The other algorithms seem to be similar in number of converged designs. For all the cases there are many unfeasible configurations due to the narrow constraints imposed to the massflow rate.

In order to have an efficient coverage of the design space, the first two classified algorithms were both selected for the generation of the ANN training DoE.

The Latin Hypercube sampling technique generates random numbers following statistical distributions. It stems from the Monte Carlo method. Considering the normal distribution of an input variable in its domain, the distribution is divided into \( n \) intervals with equal probability distribution and a random value is chosen for each interval. In this way the points are proportionally displaced following the value assumed by the probability density function (higher concentration of points where the probability density function is higher and viceversa). Hereafter the Latin hypercube sampling is intended with normal distribution.

The Sobol sampling technique is based on a quasi-random deterministic algorithm. It allows a fast and relatively uniform sampling of the design space. It is more efficient than the pure Random sampler for a reduced number of variables (typically 25) because it avoids the points clustering.

5.1.2 Neural Networks and DoE: minimizing the NMSETD

Once the algorithms have been selected, an analysis of the number of samples influence on the accuracy of the ANN-based metamodels was carried out.

The neural networks adopted in this procedure were the classical feed-forward multilayer perceptrons with a single hidden layer and supervised learning with Levenberg-Marquardt backpropagation algorithm extensively documented in literature [1, 2].

Two cases were examined:

1. a full Latin hypercube DoE;
2. a fifty-fifty Sobol and Latin hypercube DoE.

Considering \( Y \) as the generic model response and \( Y' \) as the metamodel response, the error is defined as the difference between \( Y \) and \( Y' \), being \( (Y-Y')^2 \) the squared error. If \( N \) samples are considered, there are \( Y_i \) and \( Y'_i \), for \( i=1,2,...,N \), originating \( N \) corresponding errors. In order to compare the errors, a normalization is applied. Starting from a DoE of 200 samples, with a step of 200 samples, 8 DoEs were tested for each case, until 1600 maximum samples. These DoEs were validated and used to estimate the NMSETD = "Normalized Mean Square Error on Training Data" (as defined in Eq. (4)) during the creation of metamodels using artificial neural networks.

\[
NMSETD = \left( \frac{(Y - Y')^2}{\text{scaled}} \right)
\]

Table 4: The DoE sensitivity data for 100% Latin Hypercube DoEs.

<table>
<thead>
<tr>
<th>Total DoE samples</th>
<th>Converged samples (feasible and unfeasible)</th>
<th>Neurons in the hidden layer</th>
<th>Total computational time for metamodels of ( \eta_s, m', \eta ) (hh:mm:ss)</th>
<th>NMSETD</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>170</td>
<td>4</td>
<td>00:00:27</td>
<td>2.92 E-03</td>
</tr>
<tr>
<td>400</td>
<td>347</td>
<td>8</td>
<td>00:03:30</td>
<td>2.20 E-04</td>
</tr>
<tr>
<td>600</td>
<td>507</td>
<td>12</td>
<td>00:12:00</td>
<td>1.51 E-05</td>
</tr>
<tr>
<td>800</td>
<td>681</td>
<td>16</td>
<td>00:30:07</td>
<td>1.48 E-05</td>
</tr>
<tr>
<td>1000</td>
<td>853</td>
<td>20</td>
<td>01:01:06</td>
<td>1.43 E-05</td>
</tr>
<tr>
<td>1200</td>
<td>995</td>
<td>23</td>
<td>01:39:24</td>
<td>6.88 E-06</td>
</tr>
<tr>
<td>1400</td>
<td>1179</td>
<td>27</td>
<td>02:45:12</td>
<td>4.35 E-06</td>
</tr>
<tr>
<td>1600</td>
<td>1334</td>
<td>31</td>
<td>06:57:18</td>
<td>8.31 E-06</td>
</tr>
</tbody>
</table>

The automatic generation rule for the number of hidden layer neurons implemented in ModeFRONTIER was adopted, and in table 4 and 5 the runtimes for the metamodels generation and the corresponding NMSETDs are reported. A good fitting to the training data is obtained for NMSEDTs lower than 1.00 E-04.
Table 5: The DoE sensitivity data for 50% Sobol and 50% Latin Hypercube DoEs.

<table>
<thead>
<tr>
<th>Total DoE samples</th>
<th>Converged samples (feasible and unfeasible)</th>
<th>Neurons in the hidden layer</th>
<th>Total computational time for metamodels of $\eta_{ts}$, $m'$, $P$ (hh:mm:ss)</th>
<th>NMSETD</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>172</td>
<td>4</td>
<td>00:00:28</td>
<td>8.77 E-04</td>
</tr>
<tr>
<td>400</td>
<td>344</td>
<td>8</td>
<td>00:03:09</td>
<td>2.46 E-04</td>
</tr>
<tr>
<td>600</td>
<td>519</td>
<td>12</td>
<td>00:11:12</td>
<td>1.81 E-04</td>
</tr>
<tr>
<td>800</td>
<td>692</td>
<td>16</td>
<td>00:31:53</td>
<td>8.87 E-05</td>
</tr>
<tr>
<td>1000</td>
<td>852</td>
<td>20</td>
<td>01:03:00</td>
<td>2.29 E-05</td>
</tr>
<tr>
<td>1200</td>
<td>1021</td>
<td>24</td>
<td>01:58:07</td>
<td>3.03 E-05</td>
</tr>
<tr>
<td>1400</td>
<td>1180</td>
<td>27</td>
<td>02:44:34</td>
<td>1.13 E-05</td>
</tr>
<tr>
<td>1600</td>
<td>1356</td>
<td>31</td>
<td>06:24:15</td>
<td>8.45 E-06</td>
</tr>
</tbody>
</table>

From Table 4 the 100% Latin Hypercube case seems to have a steeper lowering of the NMSETD; actually for the optimization the fifty-fifty case (50% Sobol - 50% Latin Hypercube) was chosen. The main reason is that the pure Latin Hypercube with Normal distribution tends to explore less the domain space extrema, being the Gaussian distribution of points normalized on the central value of the variable domain. With a hybrid DoE less points are normally distributed, but the quasi-random substrate tends to fill the space more uniformly. Such a decision can be crucial in finding optima that are quite distant from the original design configuration of the turbine.

The function $\eta_{ts}$ is more difficult to approximate with respect to massflow rate and power and it requires a higher number of samples and hidden layers. A logarithmic scale normalized on the lowest value can be more explanatory to understand the NMSETDs trends for $\eta_{ts}$, $m'$ and $P$ vs. the number of samples in the DoEs, as shown in Figures 6a), b) and c). From the above pictures it is shown that a full Latin Hypercube has a steeper convergence, but best results are achieved with the 50-50% Sobol / Latin Hypercube sampling near 1400 DoE samples.

Considering the above graphs, a DoE of 1400 converged 50% Sobol – 50% Latin Hypercube samples was built in order to create the response surfaces with artificial neural networks, and a further 300 sample DoE was built with the same technique to test the surrogate models by comparing the turbine performance predicted by the metamodels with those computed by the flow solver. Table 6 shows the output from ANNs in good agreement with the flow solver results, with a maximum absolute error less than 1%.
Table 6: Absolute error related data on the 300 samples test DoE.

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>Maximum absolute error %</th>
<th>Average absolute error %</th>
<th>Absolute error Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN_ηts</td>
<td>0.43</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>ANN_m'</td>
<td>0.35</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>ANN_P</td>
<td>0.67</td>
<td>0.10</td>
<td>0.12</td>
</tr>
</tbody>
</table>

5.2 Optimization run using the RSM approach

The ANNs have been introduced into the ModeFRONTIER workflow to substitute the TF code for the prediction of the turbine performance. The optimization settings were left unchanged with respect to the previous optimization run in order to have a direct comparison of the runtime and the final configuration obtained. A total time of 2 hours was needed to calculate 100000 design querying the metamodels (1/7 of the time required in the flow solver based optimization).

Also in case of the ANN-based optimization, interesting values of power and efficiency were reached after only 15000 iterations (about 16 min. runtime) but for the full convergence (optimum) all the design were necessary. The final configuration found was validated with the flow solver in order to test the accuracy of the RMS in that specific point and to compare the obtained performance with those of the reference turbine.

In table 7 the percentage error (defined in Eq. 5) between the performance predicted by ANNs and the TF solver are reported and a very good accuracy is evident from the RSM approach.

\[ \varepsilon_{\phi} = \left( \left( \frac{\phi_{\text{ANN}}}{\phi_{\text{TF}}} \right) - 1 \right) \times 100 \quad \text{with } \phi = \eta_{ts}, m' \text{ and } P \]  

(5)

Table 7: Percentage error between metamodel and flow solver responses for the optimized configuration.

<table>
<thead>
<tr>
<th>Metamodel ANN</th>
<th>Percentage error εφ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ηts</td>
<td>0.11</td>
</tr>
<tr>
<td>m'</td>
<td>-0.04</td>
</tr>
<tr>
<td>P</td>
<td>0.08</td>
</tr>
</tbody>
</table>

The following variations have been obtained with respect to the original turbine:

- ηts increased by 0.31%
- P increased by 0.50%
- m’ increased by 0.22% (satisfies the constraints)

The above values are obtained by comparing the performance from the TF code for both configurations (starting and optimized using RSM). The above results are in agreement with those from the optimization run based on the flow solver.

6. Comparison between the multistage turbine configurations obtained

A detailed investigation on the main parameters for the turbine stages has been conducted for the different configurations considered. The flow expansion through the turbine has been substantially kept constant for the obtained configurations with respect to the original one due to the parameterization and the optimization problem set up. In order to check the distribution of losses through the turbine the values of the average loss coefficient ALC (Eq. (6))

\[ ALC = \frac{P_{\text{t-inlet}} - P_{\text{t-outlet}}}{P_{\text{t-inlet}} - P_{\text{t-outlet}}} \]  

(6)

for each blade is shown in figure 9. The three configurations (starting, optimized using the TF code, optimized using the RMS) are compared: both optimized configurations show lower values for every blade row and the RMS based approach is in a very good agreement with the flow solver based approach. The above considerations are confirmed in the stage efficiency variation plotted in figure 10. Other useful parameters are the blade throat-area/pitch ratio and the stage degree of reaction (Eq. (7)):

\[ R = \frac{\Delta h_{\text{rotor}}}{\Delta h_{\text{rotor}} + \Delta h_{\text{rotor}}} \]  

(7)

In figures 11 and 12 the above quantities are reported and compared for the three turbines under investigation. It is interesting to note that the optimized configurations tend to have a more uniform variation of throat areas, being the blade pitch left unmodified in the parameterization scheme. In the optimized configurations R tends to be close
to 0.5, keeping the enthalpy drop of the stage equally distributed between stator and rotor. All the above performance for the turbine configurations have been computed with the TF code and the comparisons are therefore consistent. As previously discussed the TF code predicts the turbine performance with an error with respect to the experimental reference data and tends to underestimate the performance parameters. According to the author’s experience the TF code correctly predicts the trend in performance variations with respect to a simple blade restaggering operation as in the present case; it is therefore believed that the same performance improvement trend (original vs optimised) is to be expected in case of experimental test of the redesigned turbines. After a deeper investigation into the total pressure loss distributions for the different configurations (stator and rotor rows), the incidence losses have shown to give the most significant contribution to the overall pressure loss, with respect to the other loss components (leakage flows, secondary flows, profile). The obtained performance increase for the optimized configurations is, as expected, quite small due to the constraints on the optimisation problem (given meridional channel, blade numbers, axial chords and same aerodynamic profiles) that are nevertheless consistent with an industrial re-design of a given turbine that has an already high technological level. On the other hand the proposed approach is able to automatically give a turbine configuration with high performance (at least consistent with the current technological level) starting with a rough conceptual design of the turbomachine (i.e. taken from a simple 1D design code) with interesting timescales.

7. Conclusions
In this paper a comparison between two different optimization strategies for a multistage axial-flow turbine is proposed: the first approach is based on the flow simulation by means of a flow solver (TF code), the latter approach is based on surrogate models for the system response. Following the above approaches two optimized configurations have been obtained and both show very similar overall performance increase and (after a deeper investigation) the same trend for the main turbine stage parameters. This confirms that the RSM based optimization has been properly set up. A dramatic reduction in overall runtimes is obtained with the RSM based approach. The design optimization of multistage turbines is a typical problem that needs a high number of independent variables and, moreover, it would require a very computational demanding simulation tool other than the TF code. For the above considerations, keeping in mind the application of the design optimization to very large turbines (with many stages), the RSM approach developed in this work is considered the ideal candidate for its use to multistage industrial turbines.
Nomenclature

ALC: average blade total pressure loss coefficient
ANN: artificial neural network
$\Delta \beta$: stagger angle change for the blade section [°]
h: specific enthalpy [kJ/kg], blade height fraction (span)
m’: turbine massflow rate [kg/s]
n: turbine rotational speed [RPM]
NMSE: normalized mean squared error on training data
$\eta$: efficiency [-]
o/p: throat/pitch ratio
P: turbine power [kW]
R: stage degree of reaction
RSM: response surface method
T: temperature [K]
TF: throughflow

Subscripts/superscripts:
d: design condition
H: blade hub
is: isentropic
M: blade mid
st: stagnation
T: blade tip
TF: throughflow
ts: total to static
tt: total to total
1,2: inlet, exit turbine sections
-: mean value

References