

Automatic Aerodynamic Design of a Wind Turbine through Evolutionary Techniques

V. Díaz Casás¹⁾, F. Lopez Peña¹⁾, R. J. Duro²⁾, A. Lamas²⁾

1) Grupo de Ingeniería de Fluidos, University of Corunna, vdiaz@udc.es, flop@udc.es

2) Grupo de Sistemas Autónomos, University of Corunna, richard@udc.es,
alamas@udc.es

Abstract: *The aerodynamic design of wind turbine blades is carried out by means of evolutionary techniques within an automatic design environment. A simple, fast, and robust aerodynamic simulator is embedded in the design environment to predict the performance of any turbine blade produced as an intermediate individual of the evolutionary process. The aerodynamic simulator is based on blade element theory in which a panel method is combined with an integral boundary layer code to calculate the blade airfoils' characteristics. In order to reduce computations some simplifications have been applied and the results corrected by means of the application of neural network based approximations. Results of the simulations obtained using this technique, of the application of the automatic design procedure and of the operation of the wind turbines thus obtained are presented.*

Keywords: - Wind Turbines, Evolutionary Techniques, Automatic Design

1. INTRODUCTION

The aerodynamic design of wind turbine blades is crucial in achieving a good efficiency and thus in obtaining better economical performance. Therefore, any improvement in the aerodynamic design of wind turbines implies a significant profit increase during the 20 years of their lifetime. This paper proposes an approach to carry out this task by means of evolutionary techniques within an automatic design environment. For this purpose, a simple, fast, and robust aerodynamic simulator has been developed and embedded within a design environment. The simulator is able to predict the performance of any turbine produced as intermediate individual of the evolutionary process. The aerodynamic simulator is based on blade element theory in which an airfoil analysis method based on potential theory is combined with an integral boundary layer code in order to calculate the aerodynamic characteristics of blade airfoils when accounting for viscous effects. In order to reduce computational cost, some simplifications are contemplated in the airfoil performance estimation and the results corrected by means of the application of neural network based approximations. The evolutionary design environment approximation to the design problem is based on work carried out by other authors in the application of evolution for the design and optimization of different types of systems in other realms. Examples of applications in this line are described by Kirsch and Rozvany for the structural optimization of bridges and transmission towers [1], by Keane et al. [2], Song et al.

[3] or Ong et al. [4] in airplane and airfoil design optimization.

2. DESIGN ENVIRONMENT

The design environment used in this work has been created with the objective of achieving valid results in generic design processes. This leads to its implementation as a set of independent blocks which can be substituted and relinked depending on the particular design process under consideration and the types of simulations and evaluations it requires to achieve the reliability and characteristics expressed in the specification stage.

The basic structure of the design system comprises three blocks:

- Solution search block.
- Decision making block (evaluation with or without human participation).
- Computational distribution block.

The search stage is based on different types of evolutionary algorithms employing a decision block for the evaluation of candidate solutions. The algorithm

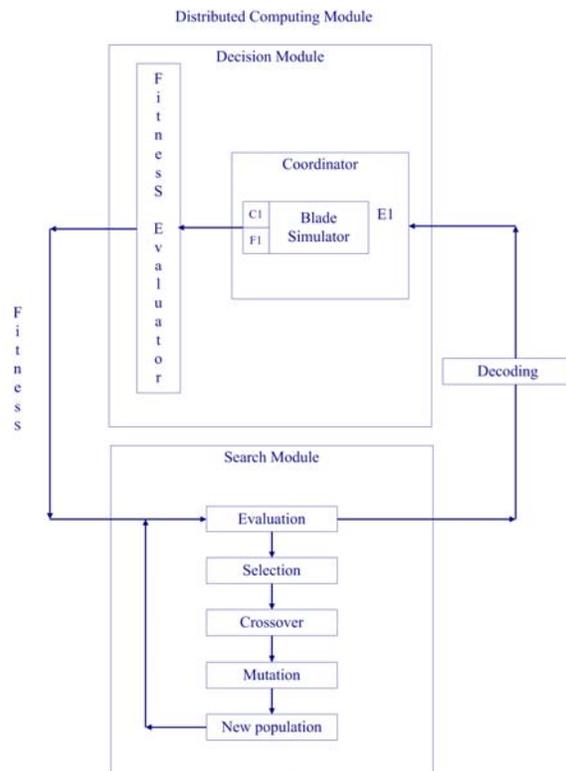


Fig.1 - Structure of the Design Environment.

initially chosen for the design of wind turbine blades is a standard macroevolutionary algorithm. The reason for this choice is that the emphasis of the work was on evaluating the operation of the system and this could be done more easily through the standard macroevolutionary algorithm. This algorithm uses a decision module based on a blade simulator that was developed ad hoc for the evaluation of each possible candidate. This simulator provides the fitness value needed for evolution to be carried out.

The evolutionary design environment used in this work is a general purpose one previously created by our research group. Both, simulators and human evaluation systems are included in the environment by means of standard interfaces allowing it to fit many different design problems. In the present case, the design is not aimed at the improvement of an initial model; on the contrary, we want to generate a completely new blade model from scratch without imposing any initial constraints. This approach makes it necessary to develop a structure that addresses all the stages of the design process. Furthermore, one of the objectives we seek is to reduce human participation as much as possible. As a general rule, we want to limit it to the mere definition of the problem. As it is common place in any design process, the initial constraints generate a space of very high, and usually variable, dimensionality but at least in the present case the search path is not conditioned by fictitious constraints which could be introduced by an arbitrary behavior of the human designer having to do with his/her level of experience and complex problem solving skills.

3. SEARCH FOR SOLUTIONS

As in any other design problem the goal here is to achieve the optimum value of a fitness function. As mentioned earlier, a macroevolutionary algorithm is used here as a search method. Initially, the evolutionary procedure implemented was a genetic algorithm based on tournament selection, random mutation and two point crossover. However the current characteristics of the fitness function led us to modify the evolutionary algorithm. Fig. 2 provides an indication of the fitness function, it displays the fitness landscape produced by fixing the values for all the genes around a good solution and testing all the possible values for two of the genes (12 and 14). The function presents discontinuities and ungrouped local maxima surrounded by flat regions produced by non-valid individuals. The presence of non valid individuals in the analysis is caused by the necessity of searching in an open region having a low level of preconditioned characteristics. In order to improve the search, a basic macro evolutionary technique has been implemented, providing for a more efficient exploration in this type of search spaces as shown by Becerra et al. [5].

The adopted search procedure is an extension of evolutionary techniques to the evolution of species. If we consider that non-adapted species are extinguished and are replaced by better adapted ones, which are created from the best individuals of the previous step. In cases, such as this one, which initially present a large number of non-valid individuals this type of algorithm could lead to a reduction in the genetic variability. To control this effect, in cases where all of the individuals are very

similar, that is, with less than 10^{-8} difference in all of their genes, one of them is replaced by a brand new random individual in order to improve the exploration of the search space.

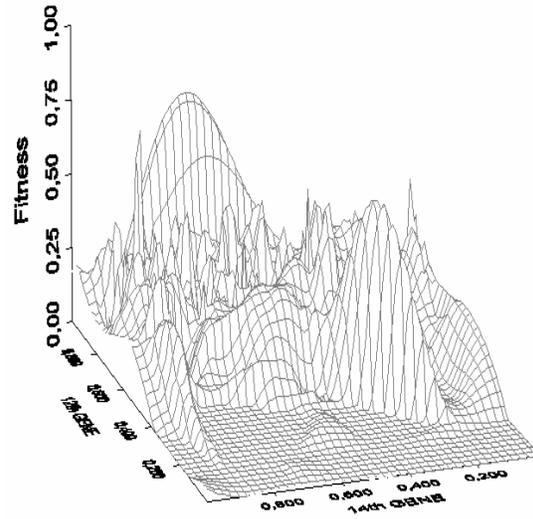


Fig.2 – Fitness function distribution considering all genes fixed except for genes 12 and 14.

4. MACROEVOLUTIONARY ALGORITHMS IN BRIEF

Here, we summarize the model proposed by Marín and Solé 0, which explains the dynamics of an ecosystem based only on the relation between species. Thus, the individuals in the population are referred to as species. They can survive or become extinct in each generation of the evolutionary process. The number of species is a constant. The relation between them is established by a matrix in which the term $W_{i,j}(t)$ represents the influence of species j on species i at time t , which is a continuous value in a given range. This influence is a measure of the difference of the relative fitness of the two species, considering the distance between both in genotypic space:

$$W_{i,j} = \frac{f(p_i) - f(p_j)}{|p_i - p_j|}$$

where $p_i = (p_i^1, \dots, p_i^d)$ is the genotype of species i , with its parameters in a d -dimensional space. f represents the fitness of each species. Thus, the influence is the difference in fitness with a normalization factor that weights the distance between the two.

Two operators are applied each generation, the selection operator and the colonization operator. The selection operator defines which species survive and which species become extinct. To determine this, the “state” of each individual is calculated as:

$$S_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^p W_{i,j}(t) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

that is, if the sum of the influences of a species relative to all the other species in the population is positive, the species survives, otherwise, it becomes extinct.

The colonization operator defines how the extinct species are replaced. The authors define a probability r to determine if a new solution p_n is generated. Otherwise exploitation of surviving solutions takes place through "colonization". One of the surviving solutions, p_b , is chosen as a base to replace the extinct solution p_i , and the new species that replaces the extinct one is attracted toward p_b , in the following manner:

$$p_i(t+1) = \begin{cases} p_b(t) + \rho\lambda(p_b(t) - p_i(t)) & \text{if } \xi > \Gamma \\ p_n & \text{if } \xi \leq \Gamma \end{cases}$$

where ξ is a random number in $[0,1]$, λ a random number in $[-1,1]$, both with uniform distribution, ρ describes a maximum radius around the surviving solution and r controls the percentage of random individuals. This parameter may act as a temperature, because it can decrease in evolutionary time so as to perform a type of simulated annealing process. That is, when the temperature is low, the randomness is low, and, consequently, there is a tendency towards increased exploitation around the surviving individuals, and reduced exploration of new species. Thus, when using a macroevolutionary algorithm one can tweak with basically two parameters. On one hand, r determines what proportion of the species are randomly generated, that is, how much exploitation or exploration we perform in a given generation. On the other, one can modify ρ and thus juggle with the size of the attractor basin around p_b , that is, it permits deciding how the exploitation is carried out. In this paper we have only played with r , and ρ has always had the value found in Marín and Solé 0 (0.5).

5. DECISION MODULE

This module includes a blade simulator able to obtain the forces and moments generated by the wind turbine blade and then calculate the resulting power generated by the turbine. To achieve the best possible simulation, a realistic CFD (computational fluid dynamics) code should be used. However, in order to reduce computational costs, a specific simplified blade simulator based on the well known blade element theory has been developed. In cases requiring a higher precision, this module could be replaced by another based on a complex CFD code, but the method presented here will not lose its generality. The present simulator is divided into two main parts; the first computes the performance of two-dimensional airfoils and the second one applies blade element theory in order to obtain the resulting forces and moments exerted on the blade. The first of these two processes, that is calculating the characteristics of the airfoil, has been divided into three phases:

- Analysis of the ideal flow around the airfoil.
- Study of the boundary layer.
- Correction, if needed, of the estimated forces values due to the detachment of the boundary layer.

The effective power produced by each blade is consequently calculated by integrating the effects of each

element's airfoil along the blade. To evaluate the characteristics of each airfoil, viscous forces and turbulence effects are initially neglected. Thus, a basic fluid dynamics principle indicates that the corresponding 2D velocity field can be obtained as the gradient of a potential function. The pressure acting on the airfoil surface is obtained by calculating the flow kinetic energy variation along the airfoil. Resulting lift and drag forces are calculated by integration of the pressure distribution along its surface.

In the previous scheme fluid viscosity is being neglected. This implies ignoring an important component of drag forces. The effects of viscous processes appearing on the blade surface can be accounted for by considering the boundary layer. An effective way to model this layer is by using a classical integral method. In our case the ones by Thwaites [7], Michel [8] and Head [9], are used in laminar, transitional and turbulent boundary layers, respectively. After that, the airfoil drag coefficient is calculated by using the Squire-Young formula [10].

The broad range of angles of attack realized in wind turbine airfoils can include large values where detachment of flow takes place. When this happens, the accuracy of potential flow theory is compromised, especially if the separated region is large. To correct the deviations from the ideal case generated in these cases, a neural network is used within our scheme. We made use of a simple multilayer perceptron neural network with 6 input neurons, two hidden layers having 8 neurons each and 2 output neurons (lift and drag coefficients). The input neurons correspond to: calculated lift coefficient, calculated drag coefficient, Reynolds number, angle of attack, separation point in the upper surface and separation point in the lower surface. These parameters were selected because lift and drag coefficients achieve near real results. They contemplate two classes of parameters: two that depend only on the case under study (attack angle and Reynolds) and four that depend on the airfoil profile and the analyzed case.

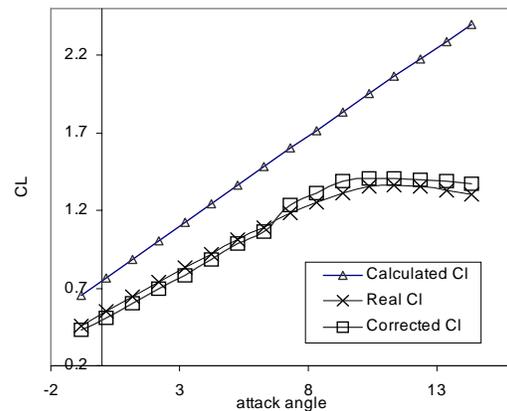


Fig. 3. Real, calculated and corrected lift coefficient for NACA 6409 airfoil. The correction is performed by an ANN.

To train the neural network, 450 real data points were employed. The data was taken from published

experiments on NACA 2410, NACA 0009, NACA 2415, NACA 2541, CLARKY, ClarkySM and DAE11 airfoils. After 1000 steps of training, an average error of around 5% is obtained; this confirms that the mathematical approach used is appropriate for our problem. The simulator implemented here achieves an adequate precision at a fraction of the computational cost required by a turbulent flow solver.

To extrapolate 2D airfoil results to 3D blade elements, it is necessary to account for the effect of fluid circulation around the blade and its wake. This effect is introduced by applying the Prandtl correction [11][12]. In addition, the attack angle of each airfoil maximizing the drag-lift ratio is calculated.

6. EXPERIMENT

A second, but important goal of this work is to develop a systematic process for automatic design. For that purpose, a general decision module with easily adjustable parameters and independent of the simulator has been created. Due to these characteristics this module can be used to solve any design problem.

The first phase when using evolutionary techniques in an automatic design, is determining the real conditions of the problem and the parameters needed to define every individual.

As a particular test problem, the environment developed is being used in designing a wind mill blade having a maximum diameter of 4.5 m. and an optimum wind speed operation condition of 7 m/s. A theoretical maximum power, called the Betz limit [13], can be easily obtained for any windmill; this is used as a control parameter for the fitness function. The Betz limit can never be achieved in reality, but it is useful as a measure of the quality of a blade by comparison.

A blade is defined by providing its distributions of chord, twist and airfoil shape along the span. In this case we have settled for the initial and final sections, considering the rest of the sections as interpolations from these. In order to define the blade sections or airfoils, we have resorted to the NACA four digit family of airfoils, which permits their definition through four parameters. The twist along the blade is established by setting the initial and final angles and the position where the mean value is achieved. The last design parameter is the angular velocity of the wind turbine, which is determined by means of the so called lambda parameter (ratio between the velocities of the blade tip and wind).

Each blade is thus encoded in the evolutionary process by a chromosome made up of fifteen genes. The range of search values is:

- NACA profiles: 0 0 0 0 to 6 6 1 9
- Angle: -20 to 40 degrees
- Initial Chord: 0 to 0.5
- Final Chord: 0 to 0.3

In this study, structural forces are not considered. However chord limitations are introduced in order to achieve the possibility of having a real blade chord distribution, for this reason, only decreasing distributions are considered.

The evolutionary technique used considered a population of 200 individuals. As commented before, the

analysis of the fitness landscape generated by the fitness function has made us resort to a basic macroevolutionary algorithm. In this method, all non-valid individuals (zero fitness individuals) are extinguished and replaced by descendents of the best couple of individuals. In order to increase the genetic variability, in each generation the worst individual is replaced by a randomly generated one. The selection is carried out through a tournament scheme among four individuals. The crossover is a simple two point crossover with a rate of 80% and the mutation is random with a rate of 0.5% for each gene, leading to a probability of an individual mutating of 7%.

To implement this evolutionary process, a cluster of 10 personal computers running a Windows XP operating system was implemented. This option was adopted because most of the commercial engineering design and numerical simulation applications run under this operating system. The objective was to make use of idle cpu time of the computers when nobody was using them. To distribute the process, MPICH2 by Argonne National Laboratory was used. It was also decided that the programs were started and closed by the distributed program.

The results presented in this work were achieved in the 65th generation, after 150 hours of calculation. Fig 3 shows the evolution maximum and of the mean value of the fitness for each generation. The fitness is defined as the ratio between the power of the wind turbine under study and the Betz power. Mechanical performance and other simplifications included in the simulator induce some errors in the numeric result, but do not affect the comparisons between blade shapes.

Fig. 4 displays the shape of the best blade obtained for the conditions stated above. This shape was achieved in the 54th generation and its efficiency is of 82.52%. The power generated by this blade is very close to the optimum for the case under consideration.

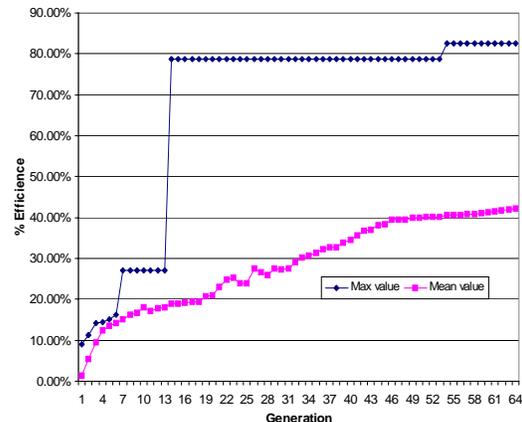


Fig. 4. Mean and maximum fitness value during evolution process.

7. CONCLUSION

An automatic design environment and an aerodynamic simulation strategy for wind-mill blades is presented. The simulator computes the characteristics of any blade profile. This is achieved by using potential flow

methods combined with boundary layer theory and correcting the results through the implementation a neural network to account for effects having to do with detachment of the flow. In a second stage the simulator computes the performance of any blade considered as an individual within an evolutionary process. This achieved by applying a blade element method.

The results obtained in a test case were very precise and clearly meet all of the design parameters imposed without having to resort to very complicated and costly non linear models of the plant within the simulator. In the tests of the design environment, taking some particular conditions into account, near optimal results for the blade were achieved.

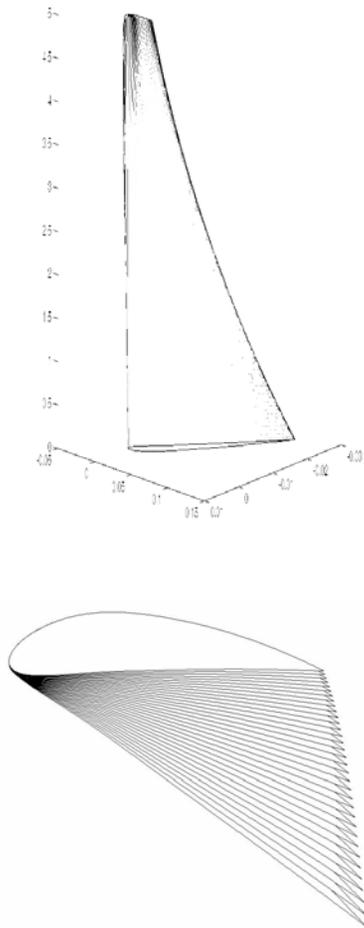


Fig. 5. 3D blade representation and top view for best individual.

This work was carried out as a first step of an integrated automatic design environment. It establishes an open basis where evolutionary parameters and simulators

can be easily exchanged to handle many different design problems.

8. ACKNOWLEDGEMENTS

This work was funded by the Xunta de Galicia under project PGIDIT03REM08E and PGIDIT04DPI166004PR.

9. REFERENCES

- [1] Kirsch, U. & Rozvany, G.I.N.: "Alternative formulations of structural optimization". *Structural Optimization*. 7, 1994, pp. 32-41
- [2] S. A. J. Keane, "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods", *J. Aircraft* 40(4), (2003). pp. 741-750
- [3] W. Song, A. J. Keane, M. H. Eres, G. E. Pound, and S. J. Cox, "Two Dimensional Airfoil Optimisation using CFD in a Grid Computing Environment" in *Proc. Euro-Par 2003, Klagenfurt (2003)*., pp. 50-57
- [4] Y.S. Ong, P.B. Nair, A.J. Keane, and K.C. Wong, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems", pp. 307-331 in *Knowledge Incorporation in Evolutionary Computation: Studies in Fuzziness and Soft Computing*, Vol. 167, ed. Yaochu Jin, Springer-Verlag, (2004).
- [5] J.A. Becerra et al. "MA vs. GA inn Low Population Evolutionary Processes with Mostly Flat Fitness Landscapes". *Proc. 6th Joint Conference on Information Sciences –2002*, Research Triangle Park. North Carolina. (2003) Pp 626-630.
- [6] Marín, J., Solé, R.V. (1999), "Macroevolutionary Algorithms: A New Optimization Method on Fitness Landscapes", *IEEE Transactions on Evolutionary Computation* 3(4):272-286
- [7] Thwaites, B. "On the momentum equation in laminar boundary layer flow". *ARC RM 2587*. 1952.
- [8] Cebeci, T., Smith, A. M. O.. "Analysis of turbulent Boundary layers". Pp 332-333, 1974
- [9] Head, M. R. "Entrainment in the turbulent boundary layers". *ARC R&M 3152*, 1958
- [10] Martin O. L. Hansen, "Aerodynamics of Wind Turbines", Science Publishers. Ltd, 2000, pp. 11-66.
- [11] John Moores, Potential Flow 2-Dimensional Vortex Panel Model: Applications to Wingmills, Master's thesis. University of Toronto. Faculty of Applied Science and Engineering. April 2003.
- [12] Jones K.D., Davids, S. and Platzauer. "Oscillation Wing Power Generation" ASME paper 99-7050 in *Proceedings of the thirds ASME/JSME Joint Fluids Engineering Conference*.
- [13] Abbot, H. and Von Doenhoff, Albert E, "Theory of Wing Sections". Dover (1949; reprinted 1958).