Neural Network Topology for Wind Turbine Analysis

A.Gaymann[1,*], *G.Schiaffini*[1,2], *M.Massini*, *F.Montomoli*[1], *A.Corsini*[2]

[1]Imperial College London, Aeronautical Engineering Department, London, SW7 2AZ, UK
[2]Universita di Roma, La Sapienza, Italy
[*]Address all correspondence to this author, email: audrey.gaymann11@imperial.ac.uk

ABSTRACT

In this work Artificial Neural Networks (ANN) are used for a multi-target optimization of the aerodynamics of a wind turbine blade. The Artificial Neural Network is used to build a meta-model of the blade, which is then optimized according to the imposed criteria. The neural networks are trained with a data set built by a series of CFD simulations and their configuration (number of neurons and layers) selected to improve performances and avoid over-fitting. The basic configuration of the airfoil is the profile S809, which is commonly used in horizontal axis wind turbines (HAWT), equipped with a Coanda jet. The design position and momentum of the jet are optimized to maximize aerodynamic efficiency and minimize the power required to activate the Coanda Jet.

KEYWORDS

Artificial Neural Network, network topology

NOMENCLATURE

ANN Artificial Neural Network
c Airfoil chord
C_D Drag Coefficient
C_L Lift Coefficient
C_μ Jet momentum
$f(x)$ Transfer function
$HAWT$ Horizontal Axis Wind Turbines
MSE Mean Square Error
NN Neural Network
t_{jet} Jet thickness
v_{inlet} Velocity at the Inlet
v_{jet} Jet input velocity
$\frac{x}{c}$ Position of the jet as fraction of the Airfoil chord
α Angle of Attack
Γ Expansion Ratio of the Jet
η Aerodynamic Efficiency
ρ_{jet} Density of Jet flow
ρ Density of incoming flow
$\omega_{i,j}$ Neuron weight
b_i Neuron bias
$\tilde{y}$ NN output
$\bar{y}$ Neuron Output
$\bar{x}$ Neuron Input

INTRODUCTION

The applications of Neural Network (NN) spans a wide variety of fields, starting from gaming algorithm, such as Alpha Go [10], to banking predictor models [4]. In recent years the application of artificial intelligence (AI) and neural networks [13, 8, 3, 5] applied to engineering design is becoming popular, and employed also in designing wind turbines[1].

The fundamental idea of a Neural Network is to mimic the behaviour of an artificial neuron, [13]. The artificial neuron receives many inputs signals and it processes these inputs via a transfer function. If the transfer function is above a threshold, then the neuron will pass a defined signal. This research led the generation of perceptron [8] as first form of Artificial Intelligence, with a two layer architecture. However only recently the computers became powerful enough to have fast back calculations, and ANN evolved to the so called Deep Learning, allowing a growing complexity in the network topology.

This paper shows the utilization of ANNs to optimize the position of a Coanda Jet along a wind turbine blade. A Coanda jet is used on wind turbine blades to enhance its circulation and, as consequence, increasing the lift. However, its position along the blade length is critical and requires an optimization process. Three design parameters are input in the NN: the angle of attack α, the position of the jet as a percentage of the blade chord, $\frac{x}{c}$, and the jet momentum C_μ. From these inputs, the Neural Networks are used to predict the wind turbine blade aerodynamics. The accuracy of the network once a design is selected, is validated using CFD simulations. An optimum position for the Coanda Jet is found through a multi-objective function. Two different size data-sets are tested to estimate the reliability of the method to build an optimum meta-model.

Test Case

This paper focuses on the wind turbine blade airfoil S809, specifically designed for HAWT. Its characteristics are widely documented, and CFD and experimental results are available in the literature [14, 12, 11, 7]. A Coanda Jet is added on the suction side [2] of this airfoil to enhance the aerodynamic lift and the stall control. The design variables are the position of the jet along the blade suction side $\frac{x}{c}$ and the jet momentum C_μ defined as :

$$C_\mu = \frac{\rho_{jet} v_{jet}^2 t_{jet}}{\frac{1}{2}\rho v^2 c} \tag{1}$$

The analysis of the airfoil has been performed in 2D, having the airfoil the same properties as a wing airfoil. The individual blade has been considered in the simulations. The following variables are used to characterize the performance of the blade airfoil for the optimization process: Lift coefficient C_L, Drag coefficient C_D, Expansion Ratio of the Jet Γ, defined as ratio of absolute pressure of Jet inlet to the absolute pressure of the undisturbed flow, and Profile efficiency η defined as [9]:

$$\eta = \frac{C_L}{C_D}. \tag{2}$$

The value of these coefficients is function of three variables: angle of attack α, jet momentum C_μ and jet position $\frac{x}{c}$. These three design variables will be the ones analyzed and modified through the optimization process, and the aerodynamics properties of the blade will be estimated by the Artificial Neura Network.

The aerodynamic performances obtained with the Coanda jet was evaluated in seven distinct configurations by varying its position along the airfoil from $0.2\frac{x}{c}$ up to $0.8\frac{x}{c}$, as seen in Fig.1. The thickness of the jet for each configuration was kept constant at $t_{jet} = 5.10^{-5}$.

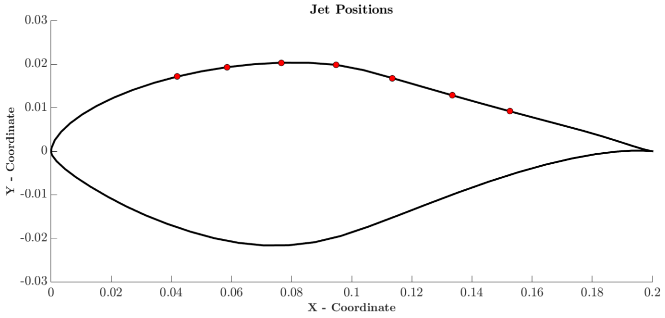

Figure 1: Coanda jet positions.

CFD model

The data used to train and test the ANN are the results of CFD simulations. Moreover, outputs of CFD simulations were employed to confirm the results obtained from the Neural Network and to validate the outputs. This section describes the CFD simulations. The mesh was defined after performing a sensitivity analysis, considering aerodynamics convergence as the mesh was refined. The final mesh selected is made of 12192 cells and 122844 nodes.

The boundary conditions are Inlet (defined with the inlet velocity), Outlet (fixed pressure value), the horizontal boundaries defined as Free Stream, the airfoil (Wall) and the Jet Velocity Inlet. Fig. 2 shows a zoom on the airfoil and the Coanda Jet. The characteristics of the boundaries are described in Table1.

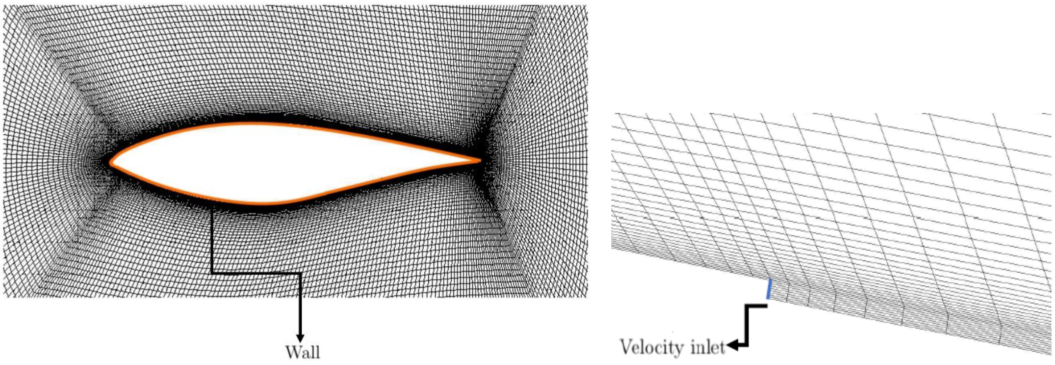

Figure 2: Boundary and mesh on the airfoil and Coanda Jet.

Steady-state simulations were performed. The air was assumed incompressible with $\rho = 1.225 kg.m^{-3}$, and dynamic viscosity $\mu = 1.8 \times 10^{-5}$. The Reynolds number of the main flow was constant, $Re = 250,000$. The heat transfer was considered negligible and the selected turbulence model was $k - \omega$ SST [6]. The variables associated to the turbulence model are turbulence intensity $I = 5\%$, turbulent and kinematic viscosity ratio $\nu_{ratio} = 10$ for both the boundaries Inlet and Jet Inlet [2]; therefore the values for k and ω are: $k = 1.5 m^2/s^2$ and $\omega = 8.3 \times 10^3 1/s$ for the boundary Inlet, and k =3.75×$10^{-3} m^2/s^2$ and ω =3.16×10^{-4} for Jet

C_μ	$v_{inlet}[m]$	$v_{jet}[m]$	$t_{jet}[m]$	$c[m]$
0.1	20	89.44	5.0^{-5}	0.2
0.07	20	74.83	5.0^{-5}	0.2
0.04	20	56.57	5.0^{-5}	0.2
0.01	20	28.28	5.0^{-5}	0.2

Table 1: Boundary Conditions numerical values.

Inlet. The height of the first cell at the wall is $10^{-5}m$, so that $y^{+} \simeq 1$. In order to validate the CFD model, the results of the numerical simulations of the basic profile S809 were compared to the experimental data available in literature [11] for several values of α.

Artificial Neural Network Architecture

In this work different strategies have been tested to build the meta-model of the blade; the best ANN performances were obtained by using three dedicated networks, one for each output parameter C$_L$, C_D,Γ. A Neural Network architecture is divided in three main blocks: the input layer, the hidden layers and the output layer. The inputs of the NN selected for the optimization process are the design variables $\frac{x}{c}$, α and C_μ. They enter the input layer and are sent to the hidden layers, which is made from multiple layers containing a different number of nodes. At each node n, the inputs of the node, $\bar{x}_i$, are multiplied by a weight $\omega_{n,i}$, then summed up; after a bias b_n is introduced at each node and added to the total sum. The total sum goes through a transfer function f, before exiting the node and then is sent to the next layer of nodes. The output $\bar{y}_n$ of the node n follows the equation:

$$\bar{y}_n = f(\sum_i \omega_{n,i}\bar{x}_i + b_n) \tag{3}$$

The transfer function form depends on the application. For the NN built in this paper, the hyperbolic tangent sigmoid transfer function and the pure linear transfer function are used. The hyperbolic tangent sigmoid function is restricted to the interval $[-1;1]$ and follows the relationship:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

The pure linear transfer function follows the equation:

$$f(x) = x \tag{5}$$

The NN are optimized to fit the outputs to the CFD results of the objective variables, which are C_L, C_D and Γ for the given inputs. Note that each objective function has its own NN. A back-propagation algorithm is used to update the weights and biases of the network following an optimization process that minimize the Mean Squared Error (MSE) cost function. Different methods are available in the literature to propagate the error to estimate the weights and biases. For this test case, a gradient based method, which relies on the calculation of the Jacobian matrix, was used.

The network topology (number of layers, nodes and type of transfer function) for this test case was chosen comparing the errors of a selection of architectures and in order to avoid problems linked to over-fitting. The latter happens when the NN complexity is too high for the information available and the response output is made to fit exactly the data. As current data can be noisy, this will impede a good fit to any data not yet observed. Over-fitting systems show very low error on training data set while error is increased for the testing set.

Data sets

To build the input data set, four values of Jet momentum C_μ were tested for 7 jet positions along $\frac{x}{c}$. Table 2 summarizes the simulations carried out for an angle of attack increment of 1°, from 0° to 18°.

$\frac{x}{c}$	v_{inlet}[m/s]	C_μ	v_{jet}[m/s]	t_{jet}[m]
0.2	20	$0.01 - 0.07 - 0,04 - 0.001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.3	20	$0.01 - 0.07 - 0,04 - 0.001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.4	20	$0.01 - 0.07 - 0,04 - 0,001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.5	20	$0.01 - 0.07 - 0,04 - 0,001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.6	20	$0.01 - 0.07 - 0,04 - 0,001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.7	20	$0.01 - 0.07 - 0,04 - 0,001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}
0.8	20	$0.01 - 0.07 - 0,04 - 0,001$	$89.44 - 74.83 - 56.57 - 28,28$	5.10^{-5}

Table 2: Table summarizing the simulations done and their characteristics

A total of 532 samples were created following these configurations and were used inside the Neural Network. 70% of the data were used to train the ANN while 30% were kept for testing it. This set will later be referenced as the complete data set. A second training data set is built with a reduced set of the original 532 simulations. The set is obtained by applying an increment of 2° the angle of attack α. The quantity of samples was consequently reduced to 280. The proportion for training and testing the ANN remains the same. This second set will be referenced as the reduced data set.

Numerical Results

The design variables jet position, $\frac{x}{c}$, jet momentum C_μ and α represents the inputs of the ANN and will be associated to a single target value at a time: C_L, C_D and Γ. The choice of training three distinct ANNs rather than a single ANN with three target variables was a consequence of an error analysis which showed that the error was smaller when the outputs were not coupled within a same ANN.

All networks were trained in a supervised manner for the two different data sets. As for their topology, that is the choice of the number of layers and neurons and the type of transfer functions to use, there is no definitive method available in the literature to determine these parameters. Only a post-processing analysis of the error can judge the quality of the architecture. To select the optimum configuration, the MSE for the different architectures was recorded and compared; as shown in Fig.3 for the target variable C_L, for both the complete and the reduced data set. A hidden layer was added every time the error associated to the testing set increased when neurons were added to the NN topology. This increase is due to an over-fitting behaviour from the NN over the training set. In the first layers the number of neurons is kept the same to have a balanced network [1]. The optimum architecture was given by the one presenting the minimum error on the testing set.

Network system for C_L fitting

For the complete data set, the best network topology to get the value of C_L is a three layers configuration with 9 neurons on the first and second layer, while the output layer is characterized by 1 neuron. For the reduced data set the architecture chosen is composed of 7 neurons on the first two layers and 1 neuron on the last layer. The difference in architecture between the two data sets is mainly explained by the fact that less data will produce simpler network architecture. Note that complex NN topology for low amount of data will lead to over-fitting problems.

The transfer function architecture for the complete data set was alternated: the first and third hidden layer use a hyperbolic tangent sigmoid function, the second hidden layer and the output layer a pure linear. The first hyperbolic tangent is necessary to allow non-linearity inside the network while the pure linear transfer function allows the transfer of data between each layers. The transfer function architecture for the reduced data set uses a hyperbolic tangent on the first and second hidden layer, while the ouput layer uses a pure linear.

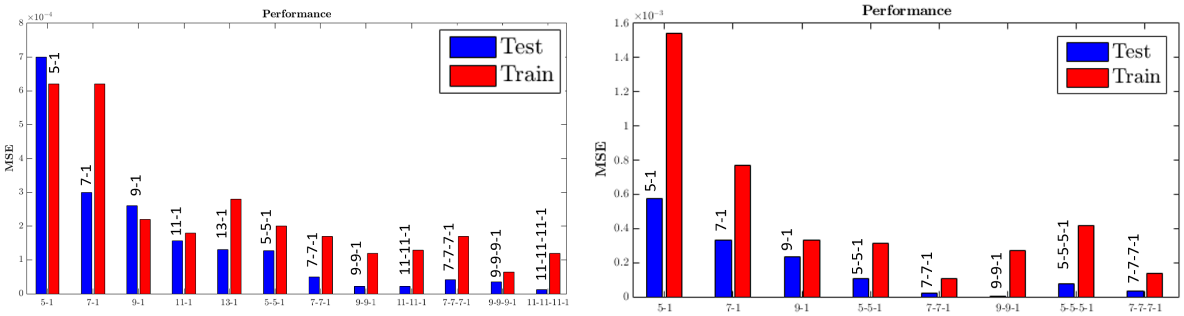

Figure 3: MSE for C_L as a function of the Network architecture. Complete data set on the left and reduced data set on the right.

Network system for C_D fitting

The same method as described in the previous subsection is used for C_D to select the optimum configuration. The optimum configuration of NN with the complete data set has 10 neurons on the first and second hidden layer, associated with a hyperbolic tangent sigmoid transfer function. The output layer has one neuron and a pure linear transfer function. The reduced data set has the same configuration as the complete data set, but the number of neurons on the first and second layer is 5.

Network system for Γ fitting

In the case of Γ, the optimum architecture for the complete dataset has 9 neurons on the first hidden layer and second hidden layer, both using hyperbolic tangent transfer function. The output layer has 1 neuron and pure linear transfer function. For the reduced dataset, the optimum configuration for the NN is found to be 7 neurons on the first and second hidden layer with a hyperbolic tangent transfer function. The ouput layer has 1 neuron and a pure linear transfer function.

Multi-objective Optimization

To build the optimum aerodynamic design considering the three design variables which are $\frac{x}{c}$, α and C_μ, a multi-objective optimization is used. A multi-objective optimization is an optimization process which meets the requirement given by the constraints and provide acceptable values for all the objectives. The objective values used here are calculated by the NN previously trained.

The target is maximizing the ratio of C_L over C_D while minimizing Γ. Minimizing Γ is minimizing the power used for the Coanda jet. During a multi-objective optimization process, the optimum solution will be obtained as a trade off between the different objectives. Indeed, an objective optimum could be reached at the expanse of the other objectives. To avoid excess penalization and to find the best trade off between the different solutions, a Pareto solution is used. Note that a Pareto Front will yield a set of optimum solutions in the space described by the objective function. A design is selected along the Pareto to fit a given criteria. The meta-model of the blade created by the ANN and its characteristics were compared to the predicted value of the CFD solver.

To solve the multi-objective optimization, a genetic algorithm is used to search the different optimum solutions. Fig.4 represents the Pareto Front for the complete and reduced data sets.

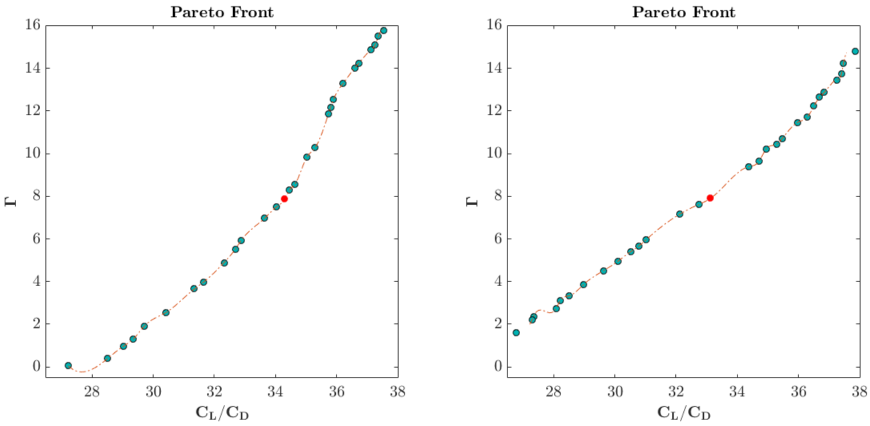

Figure 4: Pareto Front associated to the multi-objective optimization: Maximization of $\frac{C_L}{C_D}$ and minimization of Γ. Left the complete data set, right the reduced data set.

To choose a solution along the front, the ratio $\frac{C_L}{C_D}$ was normalized to a range of $[0; 1]$. The highest efficiency is given the value 1 and the lowest the value 0. The inverse methodology was applied to Γ. Both normalized values were summed and the maximum value was selected as the optimum solution. The same procedure was repeated both for the complete and reduced data set. On the Pareto Front figures, the optimum is represented by the red design point and its characteristics are reported in Table 3. It can be noticed that the two datasets produced two slightly different configurations. This discrepancy translates in different values for both efficiency and power of the jet.

Training sample	$\frac{x}{c}$	α	C_μ	C_L	C_D	$\frac{C_L}{C_D}$	Γ
532	0.63	8.58	0.006	0.88	0.00255	34.30	7.87
280	0.60	8.99	0.0056	0.89	0.00272	33.12	7.90

Table 3: Optimum design characteristics for the complete and reduced data set.

In order to confirm the values obtained, the design variables were used to validate the model with a CFD solver. The values obtained for both data sets are available in Table 4.

Training sample	$\frac{x}{c}$	α	C_μ	C_L	C_D	$\frac{C_L}{C_D}$	Γ
532	0.63	8.58	0.006	0.87	0.00267	32.58	8.46
280	0.60	8.99	0.0056	0.892	0.00272	32.79	7.96

Table 4: Optimum design CFD validations for the complete and reduced data set.

The relative error between the value obtained with the genetic algorithm and the CFD solver are summed up in Table 5.

Training sample	C_L error in %	C_D error in %	$\frac{C_L}{C_D}$ error in %	Γ error in %
532	0.864	4.71	4.99	6.4
280	0.82	0.005	1.01	1.09

Table 5: Error comparison in percentage between the NN predictions and the CFD solver predictions.

To evaluate the reliability of the configuration found through the optimization process, the error generated by the meta-model compared to the CFD solver is calculated for all the performance values. In Fig.5 the error is represented by the area in color and represents the difference between the output of the NN and the CFD simulation.

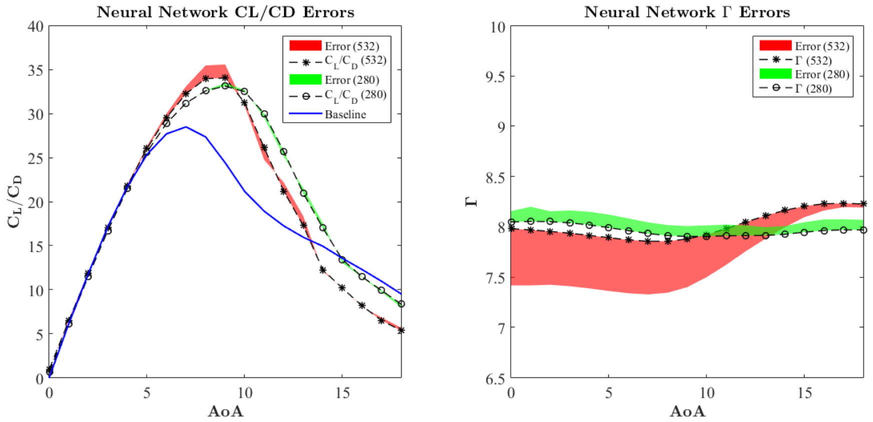

Figure 5: Error analysis for the two objectives functions vs. angle of attack, α

The error analysis shows that the reduced data set produces better results than the complete data set in terms of error. The error mean and standard deviation is given in Table 6.

The main reason for the complete data set to produce a higher error is an over-fitting problem. Its architecture includes a higher number of neurons and layers and consequently it is more sensitive to this phenomenon. However, in light of all the results available, it can be noticed that this is not a global trend for all the results as for some cases the complete data set delivers smaller error as compared the the reduced data set. Also, note that the error remains qualitatively small. The CFD simulations results can be visualized in Fig. 6.

Conclusion

From the present study it emerges that the use of neural networks for the optimization of a profile is a valid alternative to optimization methods commonly used, such as methods based

Training sample	Error	C_L	C_D	Γ	$\frac{C_L}{C_D}$
532	μ	5.2×10^{-3}	-6.1×10^{-4}	$-.037$	0.3
532	σ	1.0×10^{-2}	2.0×10^{-3}	0.2	0.7
280	μ	4.3×10^{-3}	5.5×10^{-4}	0.1	-9.0×10^{-3}
280	σ	$1.0 \times 10-2$	1.0×10^{-3}	1.5×10^{-2}	0.3

Table 6: Mean and standard deviation of error between NN prediction and CFD validation.

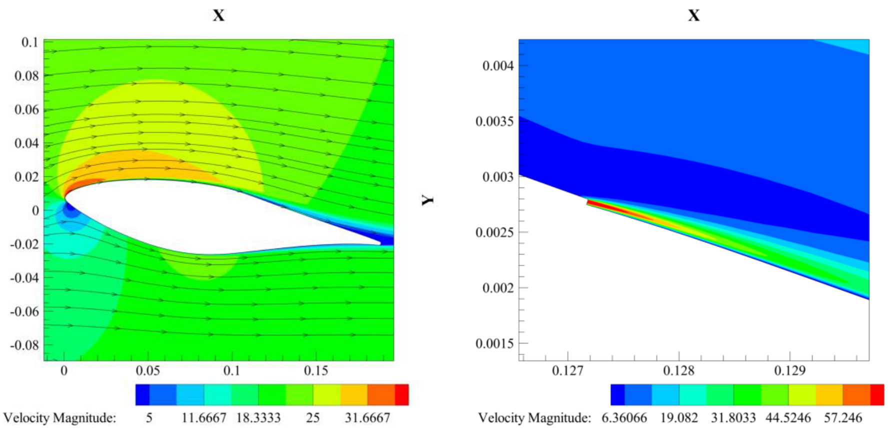

Figure 6: Complete dataset CFD validation of optimum design. On the left the velocity contour around the airfoil, on the right a zoom is done on the Coanda Jet.

on gradient-descent. Furthermore, it shows to be effective for design spaces where the multi-objective function is not characterized by a single minimum, but contains multiple local minima and maxima.

The numerical analysis shows that the ANN predicts with low error the response of the designs once compared to the CFD estimations. Problems of over-fitting inherent to the NN, have been tackled through a selection process to determine the optimum number of neurons and layers.

Slightly different configurations were obtained depending on the size of the data sets. The reason of the discrepancy lies essentially on the characteristics of the network. The CFD validation proved the validity of the response given by the NN with a low error. It was noticed during the first multi-objective optimization that the complete data set showed a considerably higher error than the reduced data set. It hints at a problem of over-fitting even though an optimization process was done to reduce its possibility. The reduced data set showed that a large amount of data is not required and a good fit from the NN is obtainable at a lesser cost on computer time resources. This study checked the validity of the method for design optimization for 2D designs. However further work needs to be done to tackle 3D optimization which will done in further work.

References

[1] Mohammad Hasan Djavareshkian Amir Latifi Bidarouni. An optimization of wind turbine airfoil possessing good stall characteristics by genetic algorithm utilizing cfd and neural network. *INTERNATIONAL JOURNAL of RENEWABLE ENERGY RESEARCH*, 2013.

[2] A. A. Jaafar et al. H. Djojodihardjo, M. F. Abdul Hamid. Computational study on the aerodynamic performance of wind turbine airfoil fitted with coandÄ jet. *Journal of Renewable Energy, vol. 2013 Article ID 839319, pp.1-17*, 2013.

[3] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[4] Milton Boyd Iebeling Kaastra. Designing a neural network for forecasting financial and economic time series. *Neurocomputing Volume 10, Issue 3, Pages 215-236*, 1996.

[5] Seymour Papert Marvin Minsky. Perceptrons. *MIT Press, 2 edition*, 1988.

[6] F.R. Menter. Two-equation eddy-viscosity turbulence models for applications. *AIAA J. 32 pp. 1598-1605*, 1994.

[7] J. M. Janiszewska R. R. Ramsay and G.M. Gregorek. Wind tunnel testing of three s809 aileron configurations for use on horizontal axis wind turbines. *NREL, pp. 1-142*, 1996.

[8] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[9] XuDong Yang ShunLei Zhang, BiFeng Song. Numerical and experimental study of the co-flow jet airfoil performance enhancement. *AIAA SciTech Forum*, 2017.

[10] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017.

[11] Dan M. Somers. Design and experimental results for the s809 airfoil. *NREL, pp. 1-103*, 1997.

[12] Warn-Gyu Park Tae-Jin Kang. Numerical investigation of active control for an s809 wind turbine airfoil. *International Journal of Precision Engineering and Manufacturing Vol. 14, No. 6, pp. 1037-1041*, 2006.

[13] Walter H. Pitts Warren S. McCulloch. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, Vol.5*, 1943.

[14] Huancheng Qu Gongnan Xie Di Zhang Yonghui Xie, Jianhui Chen and Mohammad Moshfeghi. Numerical and experimental investigation on the flow separation control of s809 airfoil with slot. *Mathematical Problems in Engineering Vol. 2013 Article ID 301748, pp. 1-14*, 2013.