Aerodynamic Coefficient Prediction of Aircraft Using Neural Network

Mailema Celestino dos Santos, mailema@gmail.com

Bento Silva de Mattos, bmattos@ita.br

Roberto da Mota Girardi, girardi@ita.br

Technological Institute of Aeronautics – Aeronautical Engineering Division – São José dos Campos – SP - Brazil

Abstract. *The present work discusses the application of neural networks for the accurate prediction of aerodynamic coefficient of airfoils and aircraft configurations. Meta-models based on neural-network are able to efficiently handle non-linear problems with a large number of variables. A methodology employing neural networks for predicting aerodynamic coefficients of generic single-airfoil configuration was developed. Basic aerodynamic coefficients are modeled depending on angle of attack, Mach number, Reynolds number, and airfoil geometry. All data are provided for a neural network, which is initially trained to learn an overall non-linear model dependent on this large number of variables. A new set of data, which can be relatively sparse, is then supplied to the network to produce a new model consistent with the previous model and the new data. The new model interpolates with high accuracy in the sparse test data points and the obtaining of a result for a generic configuration is a relatively easy and quick task. Because of this, the methodology is highly suitable to be fitted into a multi-disciplinary design and optimization framework, which makes extensive use of aerodynamic parameters to calculate performance and loads, besides other core tasks. A Multilayer Perceptrons (MLP) network was designed and employed for predicting drag polar curves of generic airfoils for a given Mach and Reynolds number variation. Airfoil geometry is modeled by polynomial functions dependent on twelve variables.*

***Keywords**: artificial neural networks, applied aerodynamics, airplane design*

1. INTRODUCTION

For centuries, the scientific approach for the understanding of physical laws was based on the construction of mathematical models. Usually solving a non-linear system of equations, the behavior of physical phenomena could then be known or estimated. Mathematical models can be used to describe the behavior of the non-linear systems, provided that initial conditions and boundary conditions are furnished. However, new simulation tools, among them neural networks, appeared and are providing new ways to predict system behavior. They represent a new computing paradigm based on the parallel architecture of the brain.

The present work describes a methodology employing artificial neural networks for the prediction of aerodynamic coefficients for any airfoil and transport airplane. There is no precise agreed definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. Neural networks are able to learn a non-linear and complex relationship among a large set of variables and for this reason are highly suited to the application under consideration.

In modern software implementations of artificial neural networks the approach inspired by biology has more or less been abandoned for a more practical approach based on statistics and signal processing. In some of these systems neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do however have in common is the principle of non-linear, distributed, parallel and local processing and adaptation. They are composed of node elements operating in parallel (Fig. 1). We can train a neural network to perform a specific function by adjusting the values of the connections (weights) between elements (Fig. 2).

The main motive for the development of the present methodology is concerned with its use in multi-disciplinary design and optimization (MDO). The aircraft industry is pushing aircraft efficiency by improving its design capacity through the development of sophisticated MDO frameworks [Lyrio et all, 2006]. MDO allows designers to incorporate relevant disciplines simultaneously enabling a very efficient aircraft configuration according to prescribed conditions fulfilling a set of constraints. In MDO process for aircraft design it is necessary to calculate aerodynamic coefficients several times. The information then obtained is used to predict the aircraft performance, stall characteristics, among other similar crucial tasks. In the majority of MDO frameworks that have been developed low-fidelity analytical expressions or the calling of CFD (Computational Fluid Dynamics) codes are carried out in order to build up polar curves. Low-fidelity analyses are conducted only in the early stages of the aircraft design and introduce a great uncertainty concerning the performance and true characteristics of the optimal configuration. Otherwise, the calling of CFD codes turns the process extremely time-consuming [Lyrio et all, 2006]. In this highlight, metamodels based on artificial neural networks have a natural application. Neural networks are able to quickly and accurately estimate the aerodynamic coefficients of any airplane configuration after they are trained. The training itself is an optimization problem concerning the minimization of the quadratic error from the variables existing in a given databank. Through the training process the synaptic weights are obtained (Fig. 3). Thus, the elaboration of the best suited neural network architecture for the prediction of aerodynamic coefficients of a generic airfoil is the concern of the present work. This task also involves the creating of a huge databank for the training of the network. A mathematical algorithm for the creation of the aerodynamic training databank was employed. It dramatically reduces the amount of data and consequently the computational effort required for the training process. If additional data is added to the databank, there is no need to retrain the neural networks every time a new project begins, since an adaptative learning skill is in practice. Thus, knowledge accumulated in past can always be recycled and expanded.

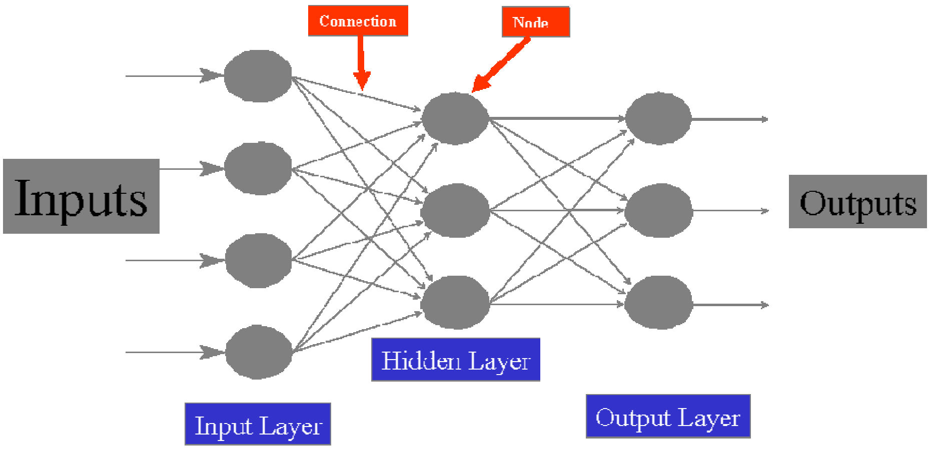

Fig. 1 – An illustration of an artificial neural network with a single intermediate layer.

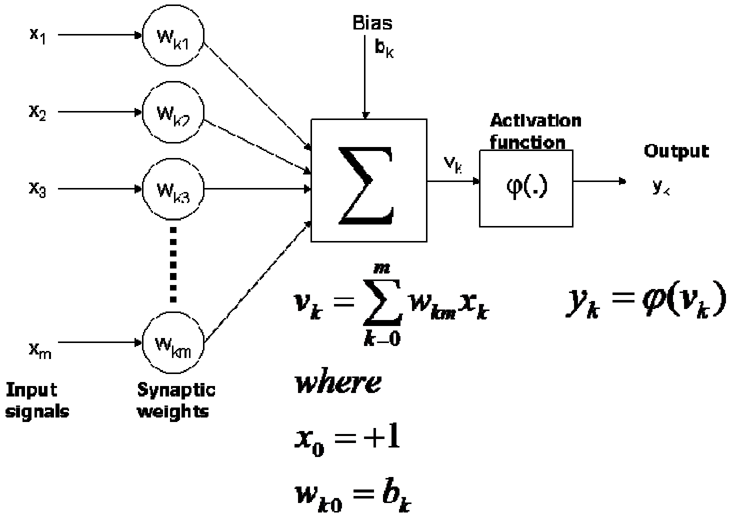

Figure 2 – Model of a neuron.

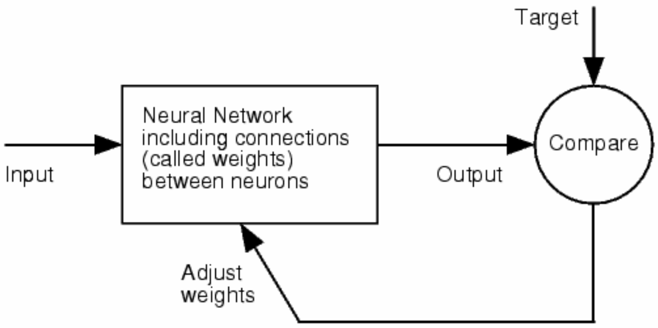

Figure 3 – The training process schematics of a neural network.

The most usual networks employed are the multi-layer perceptrons (MLP), the functional-link networks (FLN) [Curvo, 2001], and the radial basis function networks (RBF). An MLP is a network of simple neurons called perceptrons. The basic concept of a single perceptron was introduced by Rosenblatt in 1958. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Mathematically this can be written as

$$y = \varphi\left(\sum_{i=1}^{n} \varpi_i x_i + b\right) = \varphi\left(W^T X + B\right), \tag{1}$$

where W^T denotes the vector of weights, X is the vector of inputs, B is the bias and φ the activation function (log sigmoid, tan sigmoid, etc.).

In the functional link network, the hidden layer performs a functional expansion on the inputs, which gives the possibility to attach a physical meaning to the network parameters. The approximation capability of a FLN depends on the chosen set of model basis that performs the hidden layer. Provided that the set of model basis is sufficiently rich (contains sufficient high-order terms), it can be said that any continuous function can be uniformly approximated to certain accuracy. The FNL are also linear in their parameters, which means that these parameters can always be learned in the least-square sense [Curvo, 2001].

A radial basis function network is an artificial neural network which uses radial basis functions as activation functions, i.e., functions whose value depends only on the distance from the origin (Gaussian, for example).

Some neural network architectures for predicting the aerodynamic coefficients of a wing planform with considerable accuracy were already designed [Wallach, 2006]. In this work all planforms are composed of the same baseline airfoils. Therefore, the purpose of this work is to determine a neural network architecture which is able to find C_l, C_d and the stall angle for any given airfoil geometry.

2. DEVELOPMENT

Sobieczky employs mathematical expressions for the proper representation of generic airfoil geometry (shape functions). This is accomplished by the use of polynomial functions for the airfoil thickness (y_t) and camber (y_c) lines

$$y_t = a_1\sqrt{x} + a_2 x + a_3 x^2 + a_4 x^3 + a_5 x^4 \tag{2}$$

$$y_c = b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5 + b_6 x^6 \tag{3}$$

The upper- and lower-side y-coordinates at a given chord location are given by

Upper-side: $$y_u = y_t + y_c \tag{4}$$

Lower-side: $$y_l = y_t - y_c \tag{5}$$

We can determine the required airfoil geometry by applying geometric boundary conditions for the equations (1) and (2). Thus, it is possible to link the a_n , b_n coefficients in Equations (2) and (3) to the geometric variables described in Table 1.

Table 1: Geometric variables for airfoil lofting definition **Erro! A origem da referência não foi encontrada.**.

xtle	X position for LE control point
ytle	Y position for LE control point
xtth	X position for maximum thickness point
ytth	Airfoil maximum thickness
atte	TE thickness line angle
ytte	TE thickness
acle	LE camber line angle
ycth	Camber at maximum thickness
xcmc	X position for maximum camber
ycmc	Maximum camber
acte	TE camber line angle
ycte	TE camber

Abbreviations: LE = leading edge and TE = trailing edge.

Lyrio suggests restricting the geometric variables values according to boundaries listed in Table 2. This eliminates the creation of geometries that depart from the airfoil one.

Table 2: Lyrio's limits for airfoil geometric variables.

Boundary → **Variable ↓**	**Lower**	**Upper**
xtle	0.015	0.015
ytle	0.028	0.060
xtth	0.300	0.450
ytth	0.100	0.170
atte	-10.000	-4.500
ytte	0.006	0.006
acle	-7.500	5.000
ycth	-0.008	0.005
xcmc	0.700	0.800
ycmc	-0.010	0.020
acte	-15	0
ycte	0	0

The aerodynamic databank was created by using the panel code XFOIL [Drela, 1989]. XFOIL is an open source code available on the Internet. The inviscid formulation of XFOIL is a simple linear-vorticity stream function panel method. A finite trailing edge base thickness is modeled with a source panel. The equations are closed with an explicit Kutta condition. A Karman-Tsien compressibility correction was incorporated in order to enable the calculation of compressible subcritical flow. The boundary layers and wake are described with a two-equation lagged dissipation integral boundary-layer formulation and an envelope e^n transition criterion. The entire viscous solution (boundary layers and wake) is strongly interacted with the incompressible potential flow via the surface transpiration model. This permits proper calculation of limited separation regions. The drag is calculated from the wake momentum thickness far downstream. A special treatment is used for a blunt trailing edge which fairly accurately accounts for base drag. A high-resolution inviscid calculation with the default 160 panels requires seconds to execute on a Pentium IV computer. Subsequent operating points for the same airfoil but different angles of attack are obtained nearly instantaneously. Fig. 4 presents a typical graphical output of a viscous flow analysis performed with the XFOIL code [Drela, 1989].

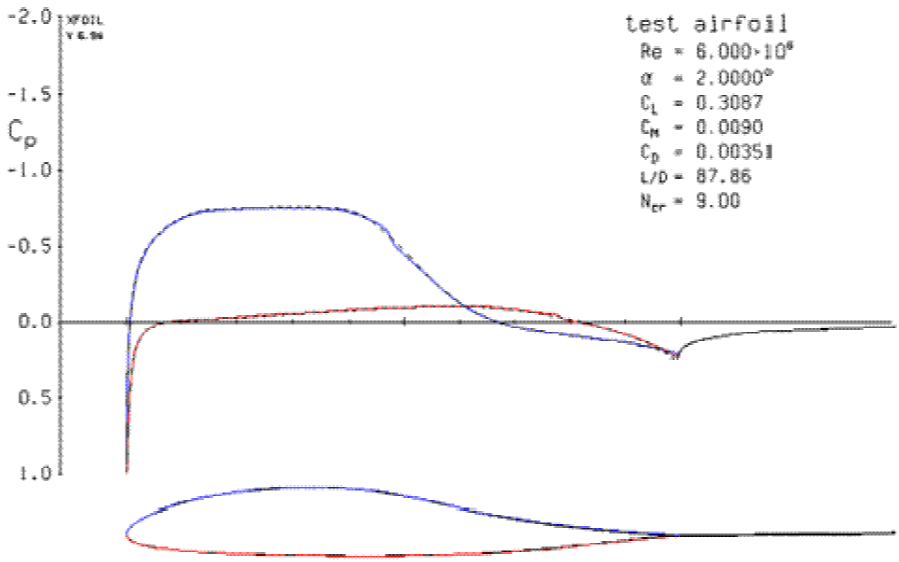

Figure 4 – Typical graph output resulted of a flow analysis performed with XFOIL.

The present methodology was implemented in Matlab® language and the MLP non-linear neural network architecture was selected as a more suited for the problem under consideration. Wallach et al. make use of this MLP architecture for predicting polar curves for the NACA 23012 airfoil with great success. In addition, Matlab® [Matlab User's Guide] furnishes a neural network toolbox that contains powerful algorithms for MLPs design and training, allowing greater flexibility and easy of use.

For the construction of the databank that is employed for the network's training, a design of experiments (DOE) technique was chosen. The fundament of this strategy was the generation of fewer data points than using passive instrumentation. In addition, the quality of the set of data is considerably higher. Latin hypercube patterns, for example, obey DOE laws and are useful when a random data sample is employed. Additionally, it is guaranteed to be relatively uniformly distributed over each dimension. As a result, the author generated Sobieczky's profile parameters through the Latin hypercube design function [Doebelin].

Provided the parameterized geometries databank is elaborated, the aerodynamic data must be furnished someway. In the present work, the airfoil aerodynamic coefficients are linked to the angle of attack. Either the data could come from wind-tunnel test or from numerical calculations or even from a combination of both sources. Here, the panel code XFOIL [Drela, 1989] with free laminar-turbulent boundary layer transition was employed for providing the aerodynamic data. The flow was set to be incompressible and the Reynolds number for all calculated coefficients is 1.0 x 10^7.

Initially, the lift (C_l) and drag (C_d) coefficients are obtained for airfoils generated with Sobieczky's profile parameters and for angle of attack ranging from -4° up to close after stall. However, neural networks are trained inside some input intervals and are able to compute the output only if the asking was between these intervals. Thus, the greatest challenge in determining drag polars is where stall takes place at different angles according to the airfoil geometry. In other words, the upper limits for the angle of attack varied for the airfoils in the training set. In order to avoid a varying input set for the angle of attack, it was parameterized as described bellow

$$\alpha_P = (\alpha + \alpha_{STALL})/(\alpha_i + \alpha_{STALL})$$

where

α_P = parameterized angle of attack; α = actual angle of attack; α_{STALL} = stall angle; α_i = lowest angle of attack (in

A MLP network architecture can be characterized by three features as follow

- number of layers,
- number of neurons in each layer,
- activation functions.

Since it is impossible to deterministically obtain all these characteristics, a trial and error procedure was employed to find them. The chosen network architecture was the best suited for the minimization of the mean square error with a normal distribution of resulting errors.

The Levenberg-Marquardt algorithm [Hagan, 1994] was employed for training the neural network. This algorithm provides a numerical solution to the mathematical problem of minimizing a function, generally nonlinear, over a space of parameters of the function. This minimization problem arises especially in least squares curve fitting. Typically an epoch of training is defined as a single presentation of all input vectors to the network. The network weights and bias are then updated according to the results of all those presentations. Training with Matlab® occurs when a previously set number of epochs is reached, or the performance goal is achieved, or even if one of the remained convergence condition occurs.

3. RESULTS

Two networks were employed for the evaluation of the present methodology: the first one was thought to predict stall characteristics of airfoils; the remained was better suited for the prediction of $C_{l\alpha}$ curves. The training database of the first network is comprised of 10,000 airfoil geometries for a fixed Mach and Reynolds numbers. The second network was trained with 2,000 geometries and its related aerodynamic data. The chosen number of Mach and Reynolds number for all data samples are 0.30 and $1.0\text{x}10^7$, respectively. The minimum and maximum values for each Sobieczky's profile parameters are the same as in Table 2. Figure 5 shows the networks training process convergence until the error between the network and the training set is below $4.0\text{x}10^{-4}$.

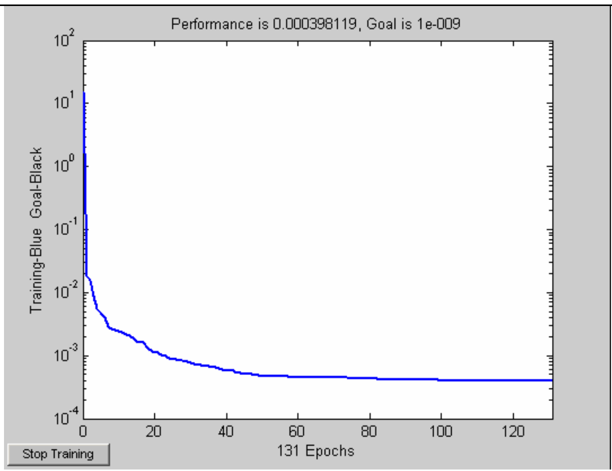

Figure 5a – Stall Training process.

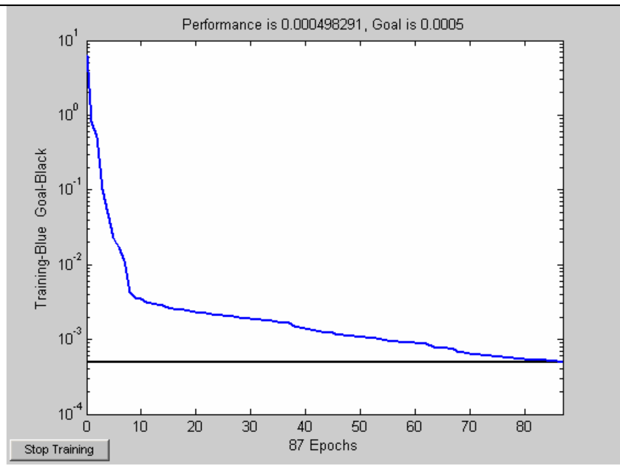

Figure 5b – *C_l x parameterized angle of attack* network training process.

The best suited neural network architecture for the prediction of the α_{max} is described in Table 3.

Table 3 - Neural Network Architecture for prediction of α_{max}.

Layer 1	**Layer 2**	**Layer 3**
50 neurons	50 neurons	One neuron
Radbas	Tan sigmoid	Pure linear

The statistical errors of the neural network for α_{max} prediction are displayed in Table 4.

Table 4 - Stall network error levels.

Average error	0.13
Standard deviation	0.96
Maximum Error	2.42
Maximum error percentage	14.25%
Minimum Error	0.02
Minimum Error Percentage	0.20%

The standard deviation of 0.96 for the α_{max} (angle of attack at stall) can be considered to be small. The figures in Table 4 can still considerably be improved if to the airfoil database new aerodynamic data is added. For the network handling *C_l vs. parameterized angle of attack* ($C_{l\alpha}$) its architecture is as follow (Table 5)

Table 5 - *C_L x parameterized angle of attack* Neural Network Architecture.

Layer 1	Layer 2	Layer 3
40 neurons	40 neurons	One neuron
Tan sigmoid	Radbas	Pure linear

The statistical characteristics of this neural network during validation were:

Table 6 - *C_L x parameterized angle of attack* network error levels.

Average error	0.0085
Standard deviation	0.0980
Maximum Error	0.3733
Maximum error percentage	110.5%
Minimum Error	0.0002
Minimum Error Percentage	0.01%

At first glance, the error levels listed in Table 6 seems to be high, the maximum error percentage reaching more than 100%. However, it corresponds to a point were the C_l is very small, close to zero. Regarding the maximum lift coefficient, typically the C_{lmax} for an aft cambered transonic airfoil is close to 2.00. For this value the standard deviation of 0.098 corresponds to an error of 4.9%. It is high due to its impact on the aircraft performance as will be demonstrated in the following paragraphs. It is very reasonable but as mentioned before all error in predicting the aerodynamic coefficients can be lowered by increasing the airfoil database. The limit in the present work was imposed by hardware (desktop computer memory, mainly) constraints.

Figure 6 shows a linear fit between expected target outputs and the results obtained by the $C_{l\alpha}$ neural network. Figure 7 shows the histogram of residuals. There is a motivating correlation between them. As presented in Table 6, the average of the errors is very close to zero and the standard deviation is small. The points with high error percentage are isolated cases.

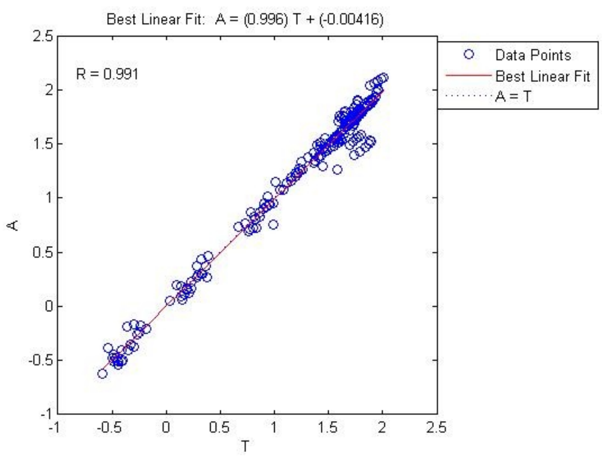

Figure 6 – Comparison between expected and predicted outputs for C_L x parameterized angle of attack.

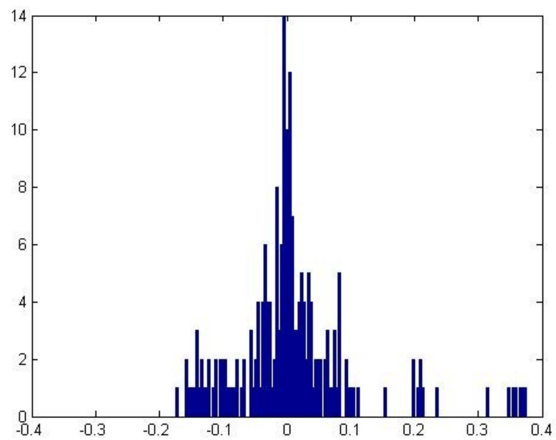

Figure 7 – Histogram of residuals – C_L x parameterized angle of attack.

When both networks are computed at once the results for *C_L x angle of attack* are displayed in Table 7.

Table 7 - C_L x angle of attack error levels.

Average error	0.0026
Standard deviation	0.11
Maximum Error	0.34
Maximum error percentage	1649%
Minimum Error	9.2×10^{-5}
Minimum Error Percentage	0.008%

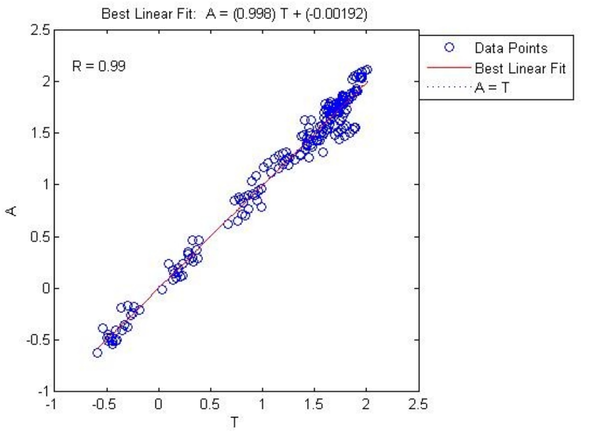

Figure 8 – Comparison between expected and predicted outputs for C_L x angle of attack.

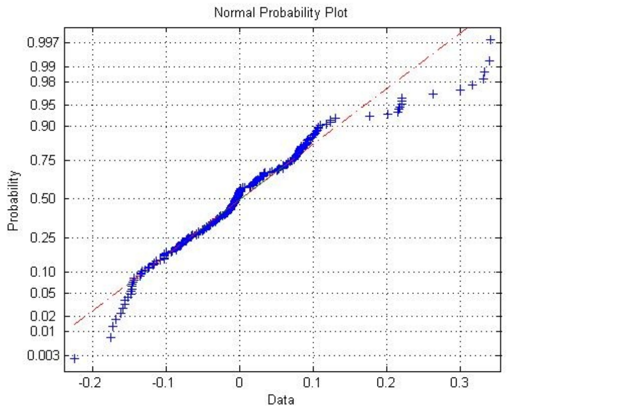

Figure 9 – Normal probability plot of residuals - C_L x angle of attack.

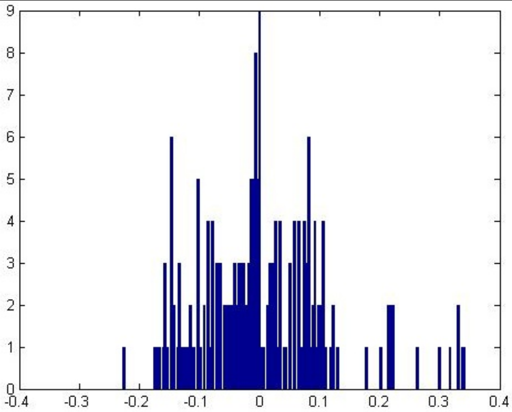

Figure 10 – Histogram of residuals – C_L x angle of attack.

Again, the maximum error level corresponds to a point where C_l is small, very close to zero. Thus, this kind of error it is not being properly filtered by the network. This error can probably be lowered by introducing symmetric airfoils into the database. Now, the standard deviation is 0.11, i.e., 5.5% for an airfoil presenting C_{lmax} of 2, which could apparently be a satisfactory value. However, a more accurate evaluation of this error can be given by an application in aircraft design. Thus, in order to evaluate the impact of this error on the design of an airliner comparable to the Fokker 100 or Embraer 190, it is computed the take-off distance of such an airplane (a low-fidelity calculation) [Kohrt, 2001].

$$d_{to} = \frac{k_{to}}{\sigma C_{Lmaxto}} \frac{m_{to}/S_w}{T_{to}/W} \tag{7}$$

Where

k_{to} is related to sea-level air density and the friction coefficient of the take-off runway, m_{to} is aircraft maximum take-off mass, S_w is the wing reference area σ is density ratio (1 at sea level), C_{Lmaxto} is the maximum lift coefficient at takeoff, T/W is the thrust-to-weight ratio @ take-off,

$C_{Lmax} = C_{Lmax\ clean} + \Delta C_{Lmax\ flap}$

Typical values for this class of airplane are

$k_{to} = 2.34\ m^3/kg$, $m_{to}/S_w = 500\ kg/m^2$, $C_{L\ max\ flap} = 0.70$, and T/W = 0.28. Thus expression (7) can be then rewritten as

$$d_{to} = \frac{4178.58}{C_{L\max clean} + 0.70} \tag{8}$$

This considering a typical value of 1.5 to $C_{Lmax\ clean}$ the takeoff distance is then computed as 1899.35 m. If the error of the two-dimensional maximum lift coefficient prediction (4.9%) could be transferred to the three-dimensional maximum lift coefficient, the takeoff distance will vary between 1966.39 and 1836.73 m. This corresponds to an uncertainty of 130 m in the take-off distance, a considerable figure. Thus, the error level of the network must be lowered.

Figures 11 and 12 show comparisons of $C_{l\alpha}$ curves predicted by the present methodology and actual ones calculated with XFOIL. The airfoil geometries used in the comparison were obtained in a random way. The A geometry presents 15.5 % maximum thickness and the B profile is 13 % thick. The predicted curve for the A case presented the greater discrepancy regarding the curve calculated by XFOIL, noticeably for C_l values below zero. In general there is a very good agreement between the predicted and calculated curves for both cases. Indeed, the C_{lmax} for both airfoils could be accurately predicted by the neural network with a very small error.

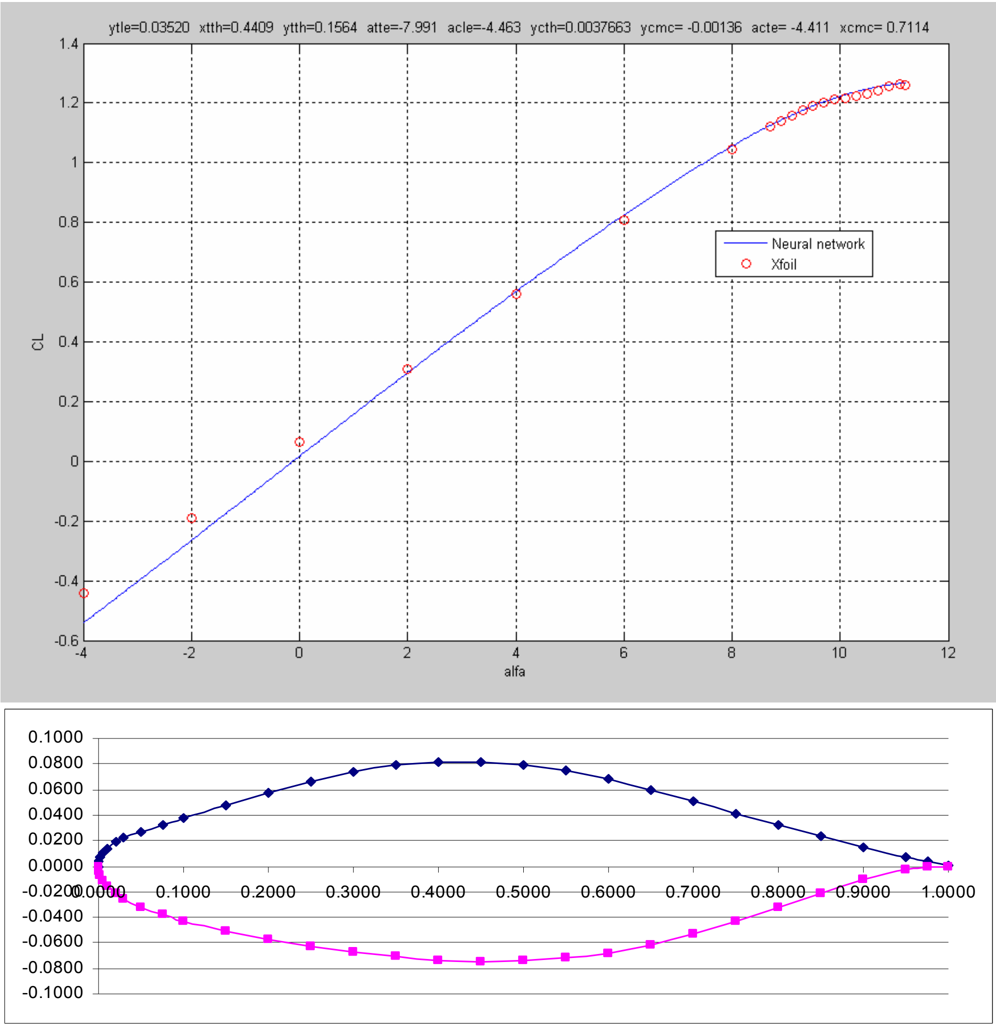

Figure 11 - Comparison between expected and predicted outputs for C_L x angle of attack (Profile A).

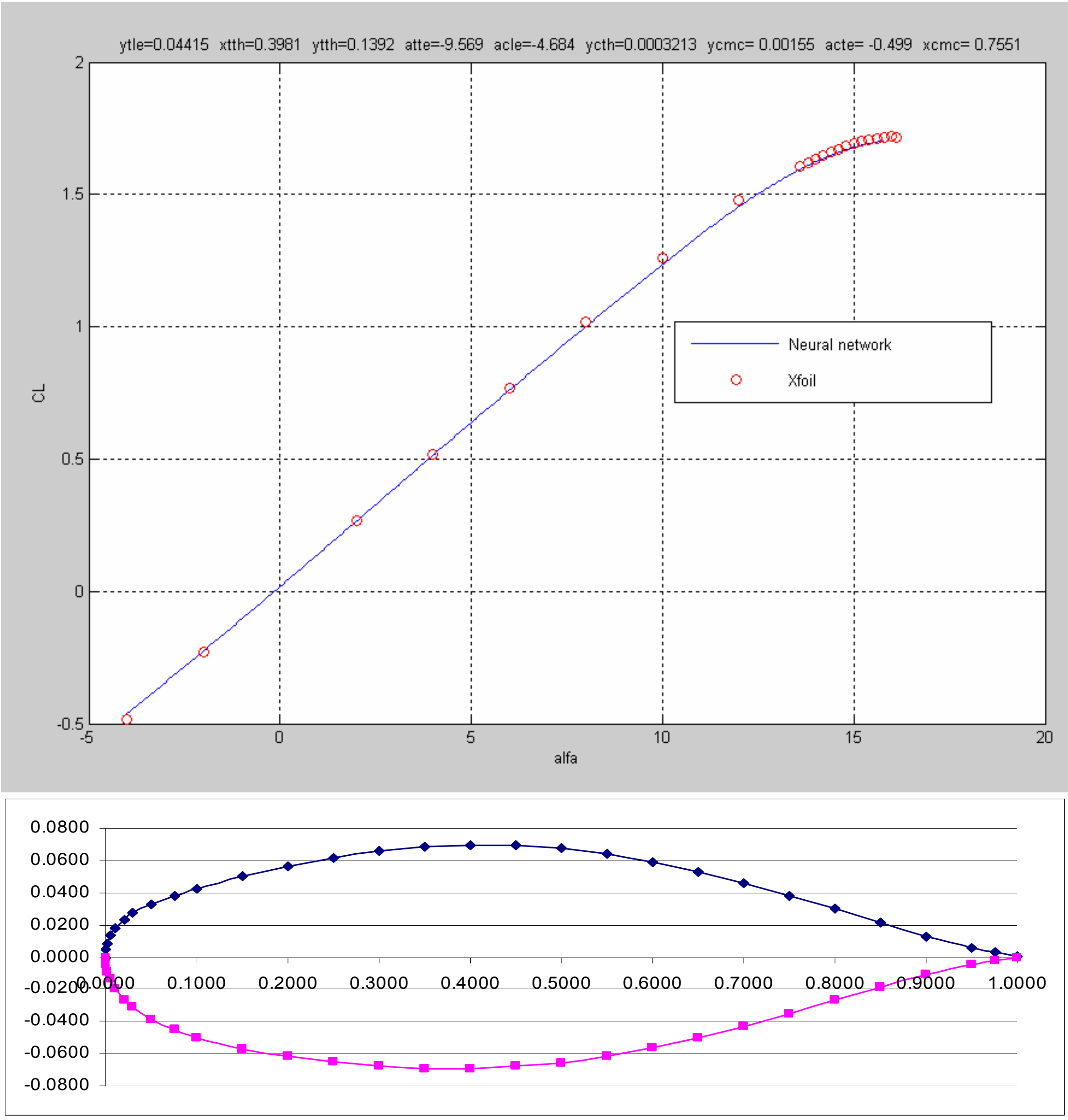

Figure 12 - Comparison between expected and predicted outputs for C_L x angle of attack (Profile B).

4. CONCLUDING REMARKS

A methodology based on artificial MLP neural network for the prediction of airfoil aerodynamic coefficients was developed and successfully applied to the subsonic regime.

The airfoil parameters limits as set up by Lyrio do not enable the proper representation of NACA airfoils. The values listed in Table 2 will be revised in order to account a broader range of airfoils to be taken into account as well as a larger number of airfoils. Symmetrical airfoils will be added to the databank in order to considerably lower the maximum error, which occur for lift coefficients close o zero.

Future work will be directed to

- Prediction of the drag coefficient (C_d) using a transonic code encompassing this way the transonic regime as well.
- Expand the prediction capabilities considering Mach and Reynolds number as input variables for the airfoil case.
- Apply the methodology for a wing-body configuration. Initially for a fixed planform and afterwards for an aircraft configuration of any wing lofting.

5. REFERENCES

Curvo, M. and Rios Neto, A.; "Aerodynamic Modeling and Stochastic Adaptive Control of High Performance Aircraft Using Artificial Neural Network and Parameter Identification", 2001. INPE – Instituto Nacional de Pesquisas Espaciais – São José dos Campos – Brazil.

Doebelin, E. "Measurement Systems – Application and Design", McGraw Hill, 2004.

Drela, M., "Xfoil: An Analysis and Design System for Low Reynolds Number Airfoils", Conference on low Reynolds Numbers Airfoil Aerodynamics, University of Notre Dame, 1989.

Fogel, David B. (2000) "Evolutionary Computation: Towards a New Philosophy of Machine Intelligence" IEEE Press, New York.

Hagan, M. T., and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.

Kohrt, M., " Flugentwurf am Beispiel einer Fokker 100," Technical Report, Fachhoschule Hamburg, September 2001.

Lyrio, J A. A., Cavalcanti, J. M. T., Mattos, B. S., Nide, G. C. R. F. J., and Paglione, P., "Complete Wing Design for Transport Aircraft," Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering -- ENCIT 2006, Braz. Soc. of Mechanical Sciences and Engineering -- ABCM, Curitiba, Brazil,- Dec. 5-8, 2006.

Matlab tutorial, R14 version 7.0.4, 2005.

Wallach, R., Mattos, B. S., and Girardi, R. M.; "Aerodynamic Coefficient Prediction of a General Transport Aircraft Using Neural Network", 2006 – ITA – Instituto Tecnológico de Aeronáutica – São José dos Campos – Brazil.

4. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.