ICAS 2006 25th International Congress of the Aeronautical Sciences

Aerodynamic Coefficient Prediction of a General Transport Aircraft Using Neural Network

Ricardo Wallach , Bento S. de Mattos, Roberto da Mota Girardi
Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil

Keywords: *applied aerodynamics, aircraft design, neural network*

Abstract

A methodology employing neural networks for predicting aerodynamic coefficients of generic aircraft was developed. Basic aerodynamic coefficients modeled as functions of angle of attack, Mach number, and Reynolds number provide data to be inputed into a neural network. In this latter case, the coefficients are also dependent on the wing geometry of the configuration. The neural network is initially trained on a relatively rich set of data from numerical simulations to learn an overall non-linear model dependent on a 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 relatively an easy quick task. Because this, the methodology is highly suited to be fitted into a multi-disciplinary design and optimization framework, which make extensively use of aerodynamic parameters to calculate performance and loads, besides other core tasks. A Multilayer Perceptrons (MLP) network was employed for predicting NACA23012 polar curves considering Reynolds number varying from 1x10^6 to 5x10^6. This two-dimensional test case was also run using a Functional-Link network in order to compare performance and accuracy from both architectures. Similarly, a two-layer network was utilized to calculate the drag coefficient of a regional twinjet of fixed geometry. For this application, the network was trained with 99 points, which represented Mach number in the 0.20 - 0.82 range. The lift coefficient in this case varied from 0 up to a determined upper limit, which decreases when the Mach number is increased. Another neural network was designed to predict the drag coefficient of a wing-fuselage combination, where the wing planform was allowed to vary. Finally, an application contemplating wing with twist was carried out. Further work will also consider drag prediction for twisted wings composed of generic airfoils.

Symbology and Abbreviations

BL	=	Abbreviation for boundary layer
C_d	=	Two-dimensional drag coefficient
C_D	=	Three-dimensional drag coefficient
C_l	=	Two-dimensional lift coefficient
C_L	=	Three-dimensional lift coefficient
M	=	Mach number
Re	=	Reynolds number
α	=	Angle of attack
AR	=	Wing aspect ratio
λ	=	Wing taper ratio
ϕ_{LE}	=	Wing leading-edge sweepback angle
Y_K	=	Wing break position referred to wingspan
CFD	=	Computational fluid dynamics
MDO	=	Multi-disciplinary design and optimization
FLN	=	Functional link network
LMS	=	Least-mean square algorithm
LHS	=	Latin hypercube sampling
RBF	=	Radial basis function
MLP	=	Multi-layer perceptron
Fig	=	Figure

1. Introduction

The present work discusses the application of neural networks for accurately predicting aerodynamic coefficients of airfoil and wing-body configurations. Meta-models based on neural-network are able to handle non-linear problems with a large amount of variables. Some test cases were carried out in order to validate the entire procedure for aerodynamic coefficient prediction. The results indicate that it is possible to improve aircraft conceptual studies[6,7], which make extensive use of aerodynamic calculations. The methodology is also suited for implementation into a multi-disciplinary aircraft design and optimization (MDO) framework. In this context, huge computer time savings without losing the required accuracy for the definition of the aircraft configuration were recorded.

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. Mathematical models can be used to describe the behavior of the non-linear systems, considering 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. Actually, neural network refers to a multifaceted representation of neural activity constituted by the essence of neurobiology, the framework of cognitive science, the art of computation, the physics of statistical mechanics, and the concepts of cybernetics. Inputs from these diverse disciplines have widened the scope of neural network modeling with the emergence of artificial neural networks and their engineering applications to pattern recognition and adaptive systems, which mimic the biological neural complex in being *trained to learn from examples*.

Neural networks are universal function estimators that contain artificial neurons. The neurons are linked by adaptive interconnections, arranged in a large parallel architecture. This arrangement produces a weighted sum of the inputs and can be *trained* to produce an accurate output for a given input. This training consists of adjusting the weights applied by the network as it sums the inputs. The power of neural networks lies in their ability to represent general relationships, and in their ability to learn these relationships directly from the data being modeled. There are essentially four broad categories of problems to which neural networks have applications: classification of patterns; function approximation; behavior prediction; and data mining.

Aircraft design is a highly complex and time-consuming task involving several strongly coupled disciplines. Considering the current highly competitive aircraft market, it has become mandatory for new designs that they must be submitted to multi-disciplinary optimization process. Aeronautical industry design objectives usually are considered in the following order: obtaining a feasible and viable configuration; to perform a robust design task; achieving an improved configuration; an optimal aircraft. Due to all those reasons, genetic algorithms has become commonplace within MDO frameworks as well as acquired widespread use in many other applications. A usual approach for MDO is concerned with the use of genetic algorithms, which require a large amount – population – of individuals, and the application of random mutations and crossing over of those individuals, for each generation of this population[4]. This approach may lead to a huge amount of different designs, which should be individually evaluated in order to segregate Pareto-optimum solutions and discard unfeasible or non-efficient results. The performance of each aircraft is evaluated based on the calculation provided by different dedicated modules for every aspect being analyzed. The analysis modules adopted in airplane MDO problems usually comprise: aerodynamics; performance; stability and control; weight and structures and, ideally, acquisition and operating costs. Aerodynamic characteristics of the population could be obtained by using analytical or semi-empirical methodologies provided by different authors[6,7]. This approach, however, presents a serious drawback: although usually qualitative results are correct, numerical results provided by such methods are highly unreliable, due to the inherent difficulty in modeling highly non-linear aerodynamic phenomena, as well as the frequent necessity of interpolating and even extrapolating relations provided by analysis of wind-tunnel experiments. Design based on these methods will never provide the level of accuracy required for MDO applied for aircraft design. In addition, in the usual approach, aerodynamics analysis of each individual in the population is done with complex and time-consuming CFD software, which can be responsible for a large amount of the total time spent in the MDO process. For these reasons, a neural-network based meta-model seems to be more suited for aircraft conceptual studies. Implementing an aerodynamics module based on neural networks has further potential advantages over the usual CFD approach:

- The calculation of the aerodynamics coefficients by CFD analysis can be only accomplished for the training set of individuals, thus dramatically reducing the amount of computational effort required for the overall design process. In this case, the coefficient values for the rest of the population are obtained much faster, by interpolating the results of the training set by means of a properly trained and validated neural network;
- The reduction in computation time provided by a neural net aerodynamic data bank could also allow increasing the size of the population under consideration and the number of generations, thus leading a broader range of available quasi-optimum solutions for the proposed problem;
- No necessity to retrain the network from scratch every time a new project begins. Neural networks can be trained accumulatively, by using the so-called *adaptative learning* algorithms. Thus, knowledge accumulated in past can always be recycled and expanded. This principle is granted by the *minimum disturbance* principle implemented in the LMS algorithm[5];
- For the cumulative training of the network, data from multiple sources, such as CFD analysis, wind tunnel and flight tests can be simultaneously used as input. In this case, special attention must be taken in order to hierarchically classify data based on its origin and reliability.

The most usual networks employed for function approximation are the multi-layer perceptrons (MLP), the functional-link Networks (FLN), and the radial basis functions networks (RBF). All these architectures are very efficient in performing data regression, and can be trained in order to output data with a desired precision[1].

Multi-layer perceptrons are formed of at least two layers of neurons. In order to the network to be able of approximating non-linear functions, it is important to have at least one hidden layer of neurons with non-linear transfer functions. The output layer of the MLP network is usually composed of neurons with linear transfer functions, in order to allow a broad range of output values. MLPs, as well as other classes of neural networks, can be fully or partially connected, and can be optimized in order to eliminate useless links, thus reducing the number of parameters in the net and allowing faster calculations.

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[3]. The approximation capability of an FLN depends on the chosen set of model basis that forms the hidden layer. Provided that the set of model bases is sufficiently rich (contains sufficient higher-order terms), it can be said that any continuous function can be uniformly approximated to certain accuracy. The FLNs are also linear in the parameters, which means that these parameters can always be learned in the least-square sense[2].

RBF neural networks are other major class of neural network model - in which the distance between the input vector and a prototype vector determines the activation of a hidden unit. RBF networks are excellent regressors, and are usually single-layered structures, which can be trained faster to the desired accuracy.

2. Network architecture

For the test cases that studied, multi-layer non-linear perceptrons networks were employed. The networks were developed using Matlab®, which contains a large number of sophisticated algorithms for training and optimization of this type of network, allowing greater flexibility in design and performance, as well as ease of implementation, compared to FLNs and RBFs set up in the same language.

As stated before, the networks studied in this paper were set up as MLPs. However, the architectures implemented for both cases are different in terms of layers arrangement, number of neurons and transfer functions.

It is not possible to set up deterministically the architecture of a non-linear network. In this case, the output layer is defined by the outputs of the problem being studied. The characteristics of the hidden layers must be defined as a compromise between the network size, the accuracy, and precision of the generated output and the training time, as well as overtraining and oscillatory behavior avoidance.

A trial and error procedure was employed for the test cases under consideration for the definition of the network architecture. The procedure algorithm compares different possibilities and chooses that with best balance between the resulting performance goal (mean square error of the outputs) and normality of distribution of the resulting errors.

2.1.1 NACA23012 airfoil

Inputs: Re; α; outputs: C_l; C_d

In a first approach, a two-layered network was employed. However, this layout was unable to resolve the acute bend present in the drag polar of the airfoil due to the free transition of the boundary layer at angle of attack close to zero lift. For this reason, a more complex design methodology was adopted. In order to approximate the existing relationship between the provided inputs and outputs, a three-layered network was adopted with the following layout:

Layer	1	2	3
Number of neurons	5	5	2
Transfer function	Tangent-sigmoid	Log-sigmoid	Pure linear

Table 1 - Network structure for the NACA 23012 Airfoil.

2.2 Transport twinjet

Inputs: Re, M, and C_L; output: C_D

The drag polar of an airplane is normally a smooth, well-behaved curve. For that reason, a simple two-layered network was able to produce excellent interpolation results in this case, and the following layout was adopted:

Layer	1	2
Number of neurons	10	1
Transfer function	Log-sigmoid	Pure linear

Table 2 - Network structure for the NACA 23012 Airfoil.

2.3 Wing-body configuration with generic wing geometry

For this case, inputs can be divided in two groups:

- Wing geometry: AR, λ, ϕ_{LE},, and Y_K.
- Flow condition: M, Re, and C_L.

The output is the drag coefficient C_D.

After many different attempts, the network architecture was defined according to **Table 3**, and presented a good compromise between accuracy of the modeled phenomenon for the training set and small oscillatory behavior for the validation data set (without overtraining).

Layer	1	2	3
Number of neurons	20	10	1
Transfer function	Tangent-sigmoid	Tangent-sigmoid	Pure linear

Table 3 - Network structure for the wing-fuselage configuration.

3. Training

Training of a neural network is the process of adjusting the network's gains and biases in order to obtain outputs as close as possible to the known target results for the training set of inputs. This array of inputs and corresponding expected outputs in known as the *training set*. The quality of the resulting outputs is evaluated by measuring the squared mean error of the network results compared to the targets, and the training algorithm is an optimization process, which tries to minimize this objective function, thus, the training procedure is also known as *Least mean square*. Moreover, the resulting errors are used as correction factors for the recalculation of the weights of the internal connections of the network, and those errors are propagated through the network from the output to the input layer, in a process so called *back propagation*. Neural networks can be trained simultaneously with a large training set (*batch* training), or accumulatively, by improving the network adding one point each time (*adaptative* training). In the cases when the training set is known before start of the process, batch training is preferable, due to its computational efficiency compared to adaptative methods. The optimization process of the LMS algorithm can be solved by several methods. For this paper, the chosen methodology was the Levenberg-Marquardt algorithm, which is an implementation of a quasi-Newton method, with variable learning rate. This algorithm ensures convergence, as in the steepest descent method, and has good performance, as does the Gauss-Newton algorithm.

3.1 NACA 23012 airfoil

The training set of inputs and outputs for the airfoil was generated using the software XFOIL[9] with the free laminar-turbulent boundary layer transition. The array consists of 125 points, generated according to the following schedule:

Parameter	Minimum value	Maximum Value
Re (x10^6)	1.00	6.00
α	-3°	18°

Table 4 - Training set for the NACA 23012 airfoil.

3.2 Transport twinjet

Training data for the network that models the drag polar of the regional jet (**Table 5**) was obtained by applying Class-II drag breakdown methodology, which consists in calculating drag coefficients for each isolated component of the airplane (e. g., wing, fuselage, etc.) and combining those components, plus interference and dynamic pressure correction factors, to obtain the total drag coefficient of the aircraft. This training set consists of 99 points of inputs and its corresponding expected outputs, arranged according to the following schedule:

Mach	0.2	0.3	0.4	0.5	0.6	0.7	0.78	0.82
Min Re (x 10^7)	1.21	1.26	1.68	1.48	1.27	1.12	1.25	1.32
Max Re (x 10^7)	1.25	1.88	2.50	3.13	3.76	4.39	4.89	5.14
Min C_L	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max C_L	1.50	1.10	0.70	0.70	0.70	0.65	0.55	0.50

Table 5 - Training set for regional jet.

3.3 Wing-body configuration with generic wing geometry

Data for training the network that models the drag coefficient of a wing-body configuration was generated using the multi-bock conservative full-potential code FPWB, which employs an approximate factorization algorithm for the marching in the pseudo-time. A total of 1300 different wing geometries and flow conditions were simulated, covering a broad range of inputs, with special attention to transonic flow conditions. This data set was generated according to a DOE technique named *Latin Hypercube Sampling*, or simply LHS algorithm[8], which tries to ensure proper spatial distribution of the data, thus maximizing coverage of the parameters' envelope (**Table 6**) with minimum correlation between samples.

	AR	λ	ϕ_{LE}	Y_K	M	Re (x10^6)	C_L
Min.	5	0.15	0°	0.25	0.20	0.5	0.30
Max.	12	0.75	45°	0.60	0.82	4	0.70

Table 6 - Training set – wing-fuselage configuration.

4 Validation

In order to validate the modeling capacities of the networks, auxiliary sets of points were generated.

4.1 NACA 23012 airfoil

The validation data for this problem consists of 500 cases (125 from training + 375 additional points), which were generated by the panel code XFOIL using the same schedule from **Section 3.1** From the analysis of this data set, results the following statistical data. **Table 7** shows the resulting error (known target minus network answer) associated with each output. As can be seen, the results for both output parameters show very small errors, with average close to zero and very small standard deviation.

4.1.1 Drag coefficient

Figure 1 shows a linear fit between expected target outputs and the results provided by the neural network. This figure shows that there is an excellent correspondence between expected and resulting outputs for the neural network.

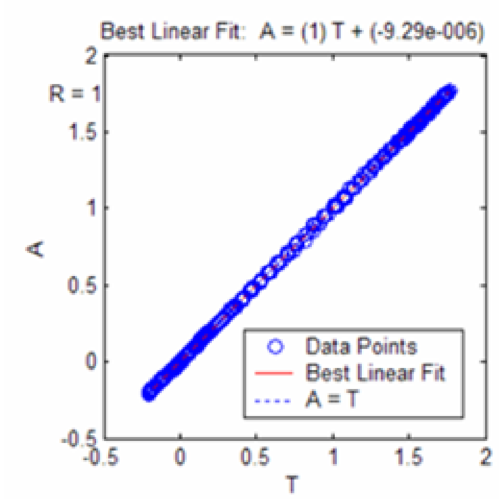

Fig. 1 - Validation fit – C_d. Comparison between expected and predicted outputs for the drag coefficient.

Associated Parameter	C_d
Average	0.00005
Standard Deviation	0.00016
Maximum	0.0016

Table 7 - Validation Results – C_d.

Figure 2 is the histogram of the errors. It shows that, as expected from the result in **Table 5**, the average of the errors is very close to zero, and has small standard deviation. **Figure 3** shows the normal probability distribution of the errors for the outputs compared to the expected values. As can be seen, the resulting errors for the drag coefficient of the airfoil show an almost normal distribution. Moreover, the network does not show signal of excess of parameters or oscillatory behavior of results. The maximum observed value of 29 drag counts occurs close to stall regions, where the total drag generally amounts 260 counts or more.

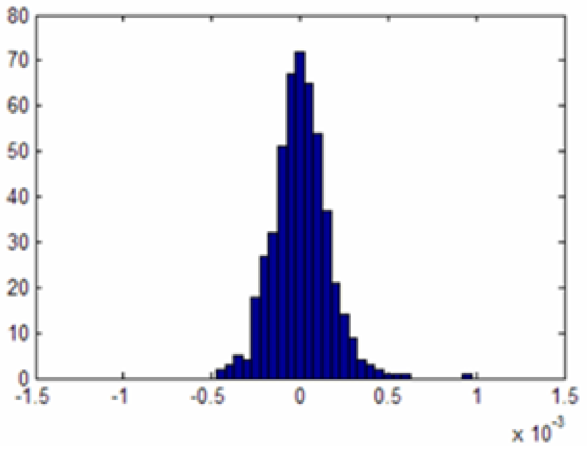

Figure 2 - Histogram of residuals – C_d. Histogram of residuals for the drag coefficient of NACA23012 airfoil.

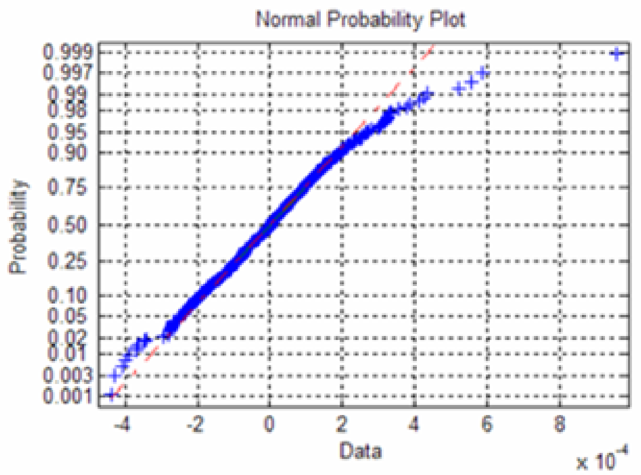

Fig. 3 - NPP of residuals – Normal probability graph of residuals for the drag coefficient.

Fig. 4 displays the comparison between expected outputs and the ones predicted by the neural network. As can be seen there is an excellent agreement between both results.

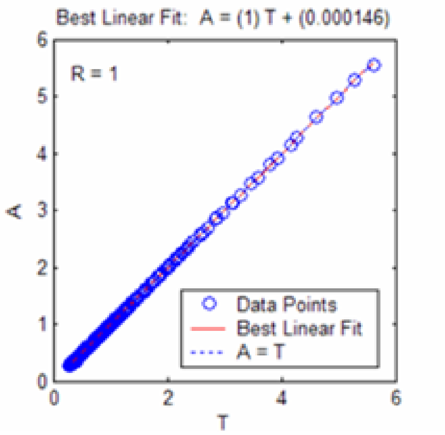

Fig. 4 - Validation fit – C_d. Comparison between expected outputs and prediction by the network.

4.2.2 Lift coefficient

For the lift coefficient, a similar statistical analysis is presented:

Associated Parameter	C_l
Average	0.0002
Standard Deviation	0.0052
Maximum	0.029

Table 8 - Validation Results – C_l.

For the lift coefficient, the histogram of the errors (**Figure 5)** shows heavy concentration of points around the average, which is also very close to zero. However, the normal probability plot of residuals (**Figure 6**) indicates that these errors do not appear to follow a Gaussian distribution. In this case, it is important to notice that statistical assumptions based on normal probability distributions cannot be stated. This fact does not invalidate the results, as can be seen in **Figure 4**, since there is still an excellent correlation between expected results and the neural network's answers for the corresponding inputs. As in the case of the drag coefficient, there is no indication of oscillatory behavior of the results.

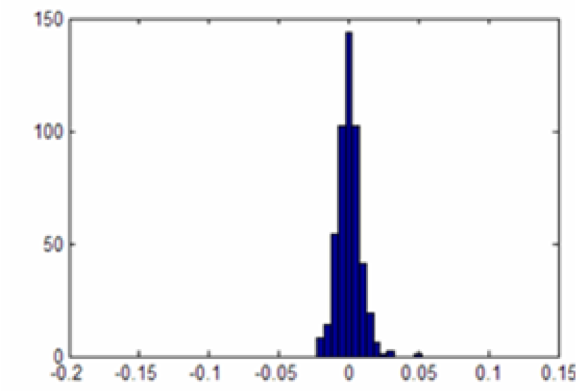

Fig. 5 - Histogram of residuals for the C_l – Histogram of residuals for the lift coefficient of NACA23012 airfoil.

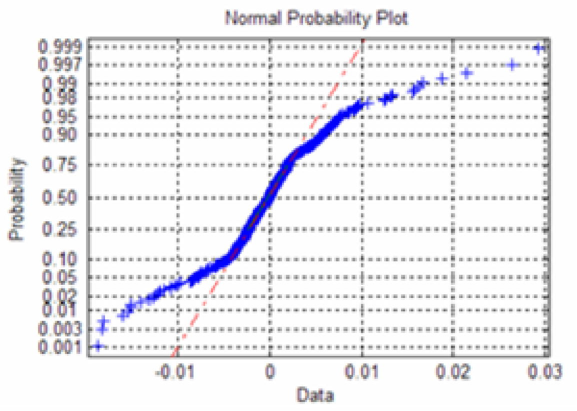

Fig. 6 - NPP of residuals – C_l. Normal probability plot of residuals for the lift coefficient of NACA23012 airfoil.

4.2 Transport twinjet

Validation data for the aircraft was also generated using a class-II drag polar generation approach. It consists of 396 points (99 from learning + 297). **Table 9** shows statistical data associated with the error relating the expected outputs of the net to the corresponding answers. In this case, the only output provided by the network is the drag coefficient of the complete aircraft. **Figure 7** shows excellent fit relating the expected outputs and the answers produced by the network.

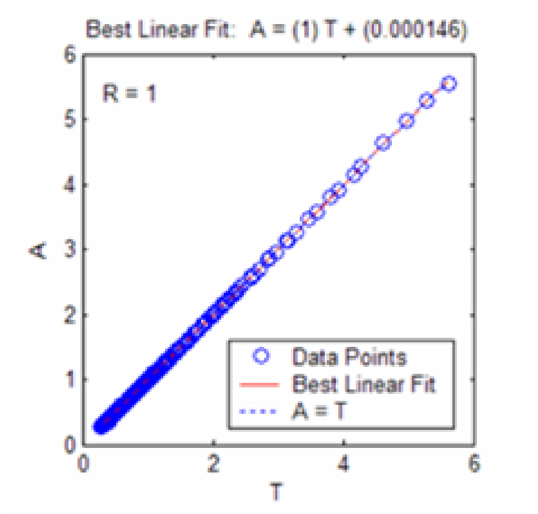

Fig. 7 – C_d Validation fit — Comparison between expected outputs and the predicted values by the neural network.

Associated Parameter	C_D
Average	-0.0000007
Standard Deviation	0.00006
Maximum	0.0007

Table 9 - Validation Results – C_D.

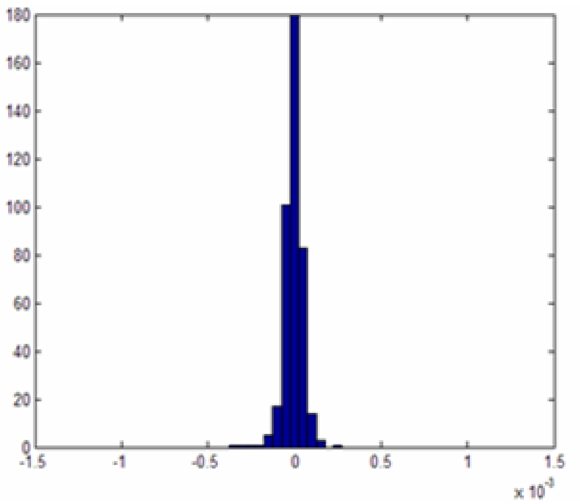

Fig. 8 - Histogram of residuals – C_d. Histogram of residuals for the drag coefficient of NACA23012 airfoil.

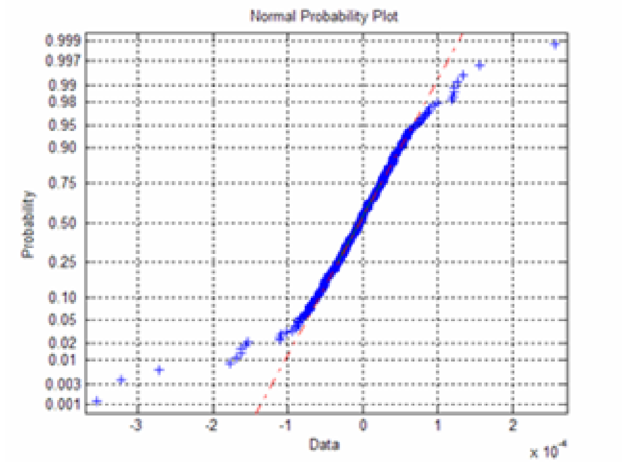

Fig. 9 - NPP of residuals - Residuals Normal probability graph of the drag coefficient.

Figure 8 shows the histogram of errors for the results produced by this network compared to the expected outputs. As in the case of the airfoil, the average of errors is very close to zero and the standard deviation of these errors is very small, actually less than 1 *drag count*. In addition, **Figure 9** shows that these errors closely follow a Gaussian distribution, so it is proper to say that more than 99% of the results outputed by the network for the validation data set have associated errors smaller than 2 *drag counts* (three standard deviations).

4.3 Wing-body configuration with variable wing geometry

Information used in the validation of the network that models the drag coefficient of the wing-body configuration was generated using the same method of the training set. This data set consists of 1550 cases, covering the same parameters schedule of the training set (**Table 6**).

Associated Parameter	C_D
Average	4.02e-06
Standard Deviation	0.00016
Maximum	0.00177

Table 10 - Validation Results – C_D.

Statistical data associated to the residuals resulting of comparison between the C_D generated by CFD analysis with FPWB code and its corresponding values as modeled by the proposed neural network can be found in **Table 10**. It shows that the mean of the residuals is close to zero, and that the maximum error found for the validation data set was of nearly 18 *drag counts*.

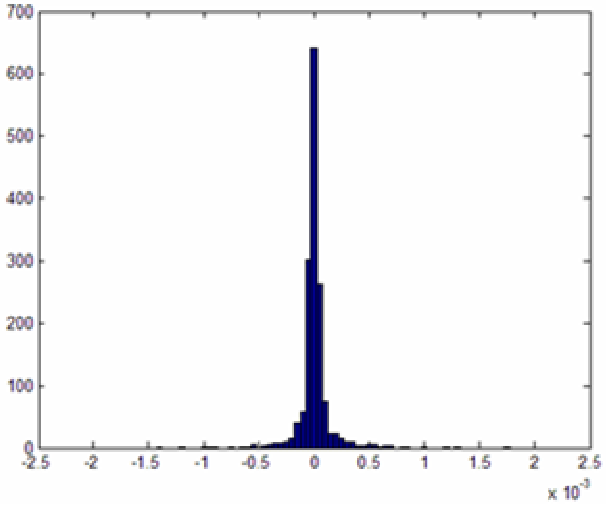

Fig. 10 - Histogram of residuals – C_d. Histogram of residuals of the drag coefficient for the wing-body configuration.

Figure 10 shows the histogram plot of the errors associated with the C_D of the wing-body configuration with generic wing geometry, loading and flow condition. **Figure 11** shows the fit between the target values of the drag coefficient (validation set only – generated with CFD) and the corresponding outputs of the neural network. As can be seen in this graph, there is an excellent correlation. **Figure 12** is the normal probability plot of the residuals. It shows that the resulting distribution of errors cannot be considered normal. However, due to the good results of the fitting in **Figure 11** and the overall low value of errors shown in **Figure 13**, which indicates that more than 97% of the points have associated errors smaller than 5 drag counts, the resulting meta-model can be considered to be very good, specially for aircraft conceptual studies.

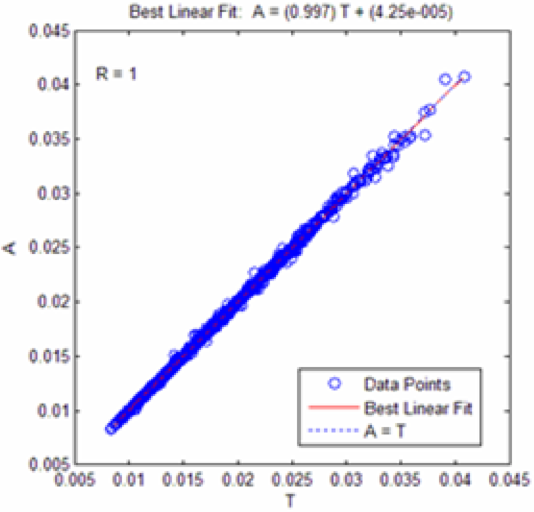

Figure 11 - Validation fit – C_D. Comparison between expected outputs and network answers for the drag coefficient of the wing-body configuration.

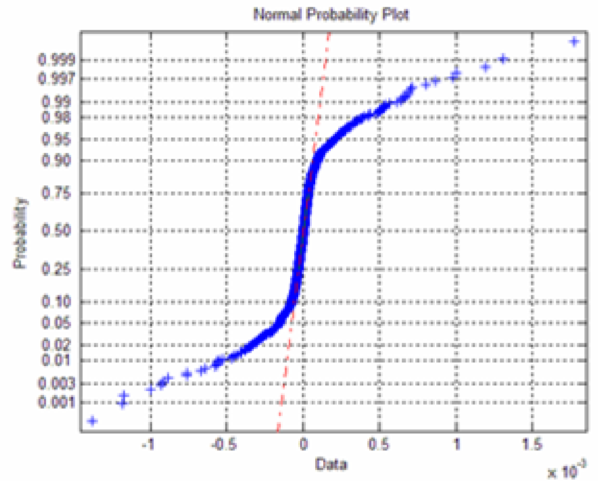

Figure 12 - NPP of residuals – C_D. Normal probability plot of residuals for the drag coefficient of the wing-body configuration.

5. Simulation examples

In order to verify the modeling capacities of the networks, some drag polar and drag coefficient curves were generated, with input data that was not used for the learning processes. For the NACA 23012 airfoil, C_l vs α curves were also generated, and compared to the corresponding data produced by the XFOIL code. For the wing-body configuration, testing points were also generating with FPWB28.

5.1 NACA 23012 airfoil

Drag polars for the NACA 230122 airfoil were calculated for Reynolds numbers of 1.5 x 10^6, 2.5 x 10^6, and 4.5 x 10^6. Alternatively, for Re=2.5 x 10^6, the results are compared to that obtained by a properly trained functional-link network stated in terms of 34 functions connected by 7th order polynomials (**Figs 14. to 17**). **Figures 16** and **17** compare the results of the MLP and FL networks, and also a direct comparison between them and the simulation outputs produced by XFOIL.

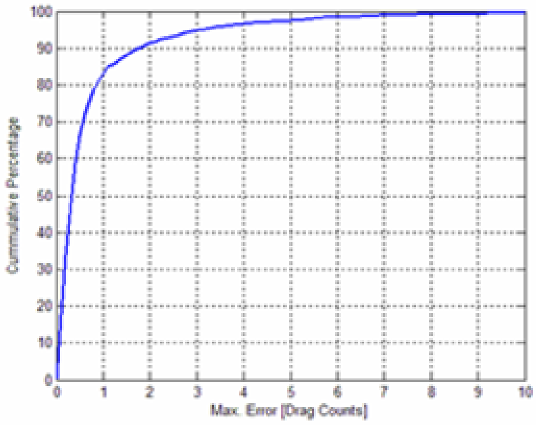

Figure 13 - Cumulative probability of errors – C_D. Percentage of points with error smaller or equal than a threshold for the wing-body configuration.

From **Figure 16** can be inferred that both neural networks can resolve the lift curve of the airfoil with excellent accuracy, even for the prediction of the maximum C_l, a highly non-linear characteristic of this airfoil. From **Figure 17**, however, it is clear that the MLP net has better capacity of capture the free BL transition jump in the drag polar when compared to the FLN.

5.1.1 Results for Re=1.5x10^6 and Mach 0.20

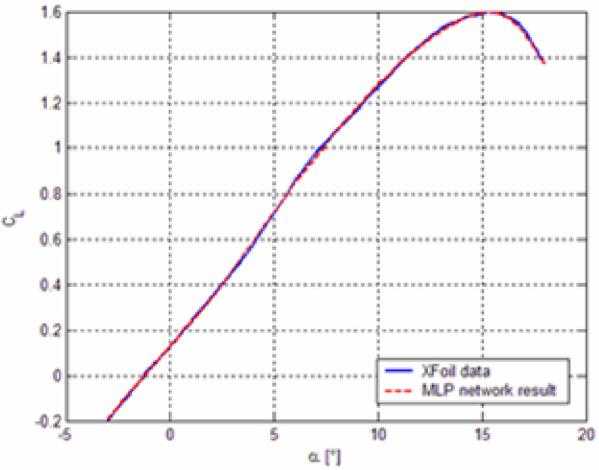

Figure 14 - C_l vs. α for NACA 23012. Re=1.5 x 10^6 and Mach number of 0.20.

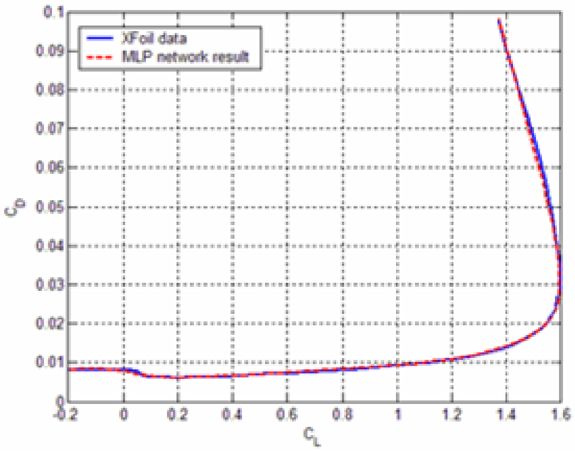

Fig. 15 - C_d vs. C_l for NACA 23012. Re = 1.5 x 10^6, Mach = 0.20, natural BL transition.

5.1.2 Results for Re=2.5 x 10^6

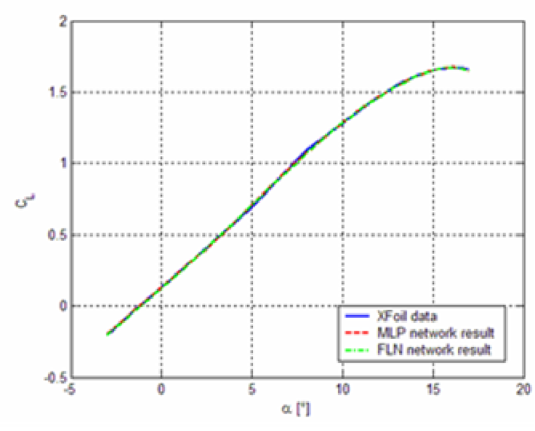

Figure 16. C_l vs. α for NACA 23012. Re = 2.5x10^6 and Mach = 0.20.

For a large of simulations involving different Reynolds number, the MLP network was able to predict the maximum lift coefficient for the NACA 23012 with error lower than 0.3 %. It would be very interesting to investigate how efficient the MLP network is able to capture the post-stall behavior.

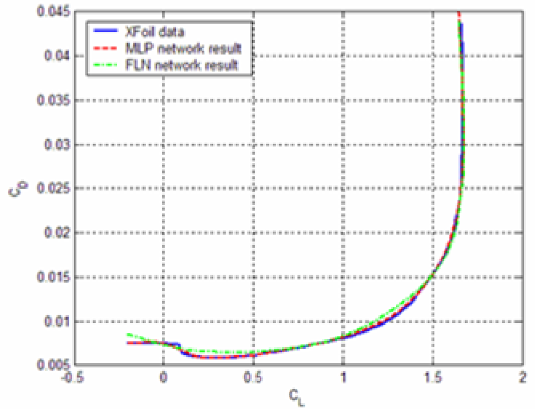

Fig. 17 – C_d vs. C_l for NACA23012. Re = 2.5 x 10^6, Mach = 0.20, and natural BL transition.

5.1.3 Results for Re=4.5 x10^6

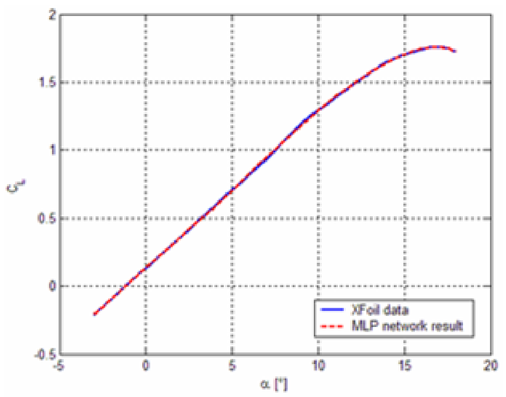

Fig. 18 - C_l vs α for NACA 23012. Re = 4.5 x 10^6 M = 0.20.

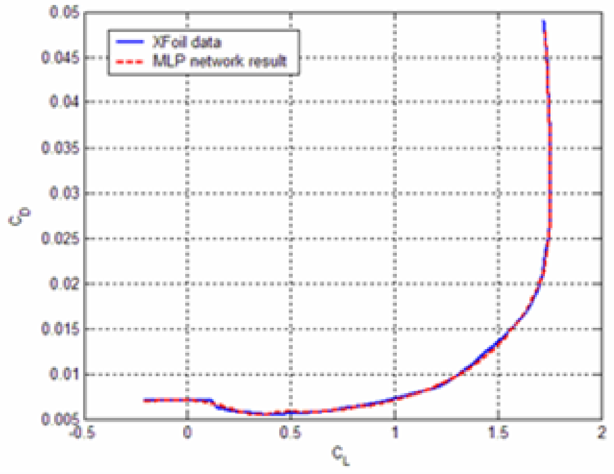

Fig. 19 - C_d vs. C_l for NACA 23012 . Re=4.5 x 10^6, Mach = 0.20, natural BL transition.

5.2 Twinjet airliner

In order to demonstrate the interpolation capabilities of the neural network drag polars of a twinjet airliner were estimated. The polars presented in this section were obtained with combinations of Reynolds and Mach number values that were not used in the network training process. A comparison to a Class-II methodology estimation for the same input can be seen in **Fig. 20**, The largest difference to the Class-II methodology was of 3 drag counts (**Figure 22**, @ C_L=0).

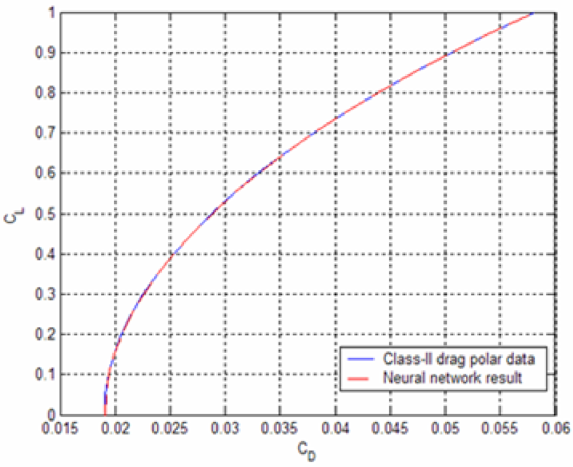

Fig. 20 - Drag polar for the regional jet test case. Mach of 0.35 and Re = 1.75 x 10^7.

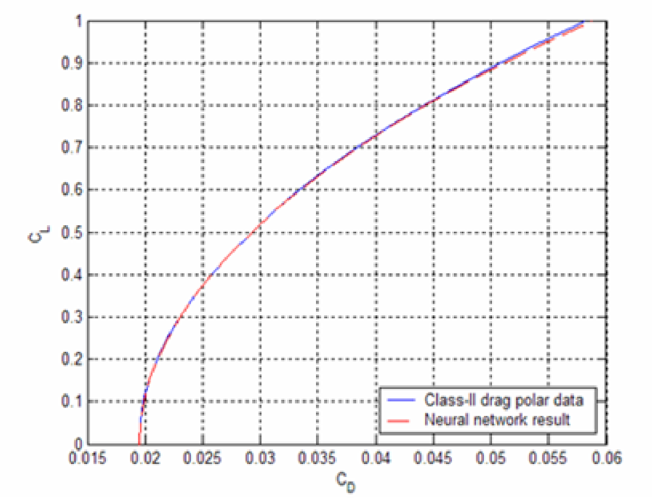

Fig. 21 - Drag polar for the regional jet test case. Mach = 0.55 and Re = 2.00 x 10^7.

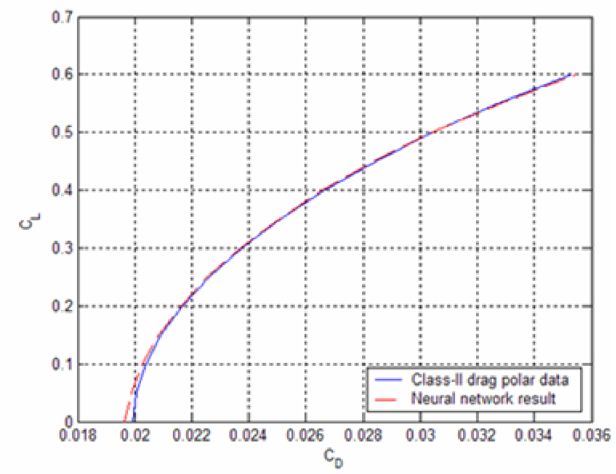

Fig. 22 - Drag polar for the regional jet test case. Mach = 0.75 and Re=2.5 x 10^7.

5.3 Wing-body configuration with generic wing geometry

The following figures show results for the simulation of the effect of changing the geometry of the wing and the flow over it, with a constant geometry fuselage. In each figure, the parameter indicated in the x-axis changes, and all the other remained six input variables are kept constant, with their corresponding values indicated in the title of the chart.

Figures 23 to **29** show the effects of the wing geometry and flow condition variation on the drag coefficient. The wing planform for these cases are more related to a typical transonic twinjet configuration. **Figures 30** to **36** show the effects of geometry and flow variation on the drag coefficient for a configuration that is resembles typical regional turboprop airplane geometries. **Figure 37** shows the parameters adopted in the description of the wing geometry.

6. Twisted wing

Simulations for a configuration with a twisted wing were also carried out in the present work. However, the twist angle that was employed is that of a trapezoidal reference wing (**Fig. 37**). Again, the network was trained with aerodynamic parameters calculated by the full-potential code FPWB. **Figs. 38-43** display some drag coefficient predictions performed by the MLP neural network. The values outputted by the network are compared with that ones obtained with the code FPWB. **Fig. 38** reveals that the MLP network was able to accurately calculate the C_D for wings of any aspect ratio with 4 degrees twist. The drag-divergence behavior was well captured by the MLP netwok for a high aspect-ratio wing with 4 degrees of twist angle (**Fig. 40**). However, the MLP output for a wing with AR=9 presenting different sweepback angles must be improved (**Fig. 43**).

7. Concluding remarks

From the results presented in the previous sections, it is possible to conclude that the implemented networks have excellent capacities for modeling the studied aerodynamic characteristics of the NACA 23012 airfoil, the complete airplane with fixed geometry and the wing-body configuration with generic wing geometry. The simulations showed that the relatively simple network structures implemented were able to resolve the highly non-linear behavior of the phenomena under consideration, even for the high degree of precision required for applications in multi-disciplinary optimization design problems. This reveals that are room for a growth in the number of variable, necessary to handle realistic aircraft configurations.

The comparison of the drag polars of the NACA 23012 airfoil generated with the 7th order polynomial FLN and the MLP networks showed that the later are better suited for the solution of problems with pronounced non-linear characteristics. Thus, the results presented in this paper indicate that the MLP network architecture is probably adequate for the work of predicting the drag coefficient of an aircraft based on its geometry and flight condition, conclusion that is supported by the excellent results obtained for the wing-body configuration where the wing geometry was allowed o vary.

The MLP network uses the lift coefficient as input for the calculation of the drag coefficient. However, the flow around the configuration with twisted wing is strongly dependent on the local angle of attack, which in turn is dependent in first instance on the angle of attack of the flow. Thus, further work will set up the training employing the angle of attack as input variable. The concept of the reference wing is an attempt to simplify calculation of more complex wings. It is also utilized to obtain coefficients from measured or calculated values of aerodynamics parameters. However, a neural network is able to easily handle complex systems with a large number of variables.

For this reason, the reference wing will not take part in future simulations with the MLP neural network.

Work under development will also consider wings with generic airfoils, i.e., the network will be able to predict drag for wing composed of any airfoil and with planform of any shape.

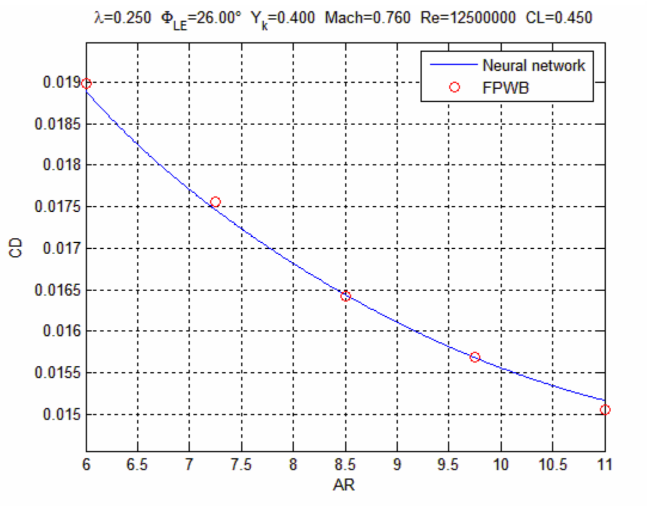

Fig. 23 - C_D variation on AR.

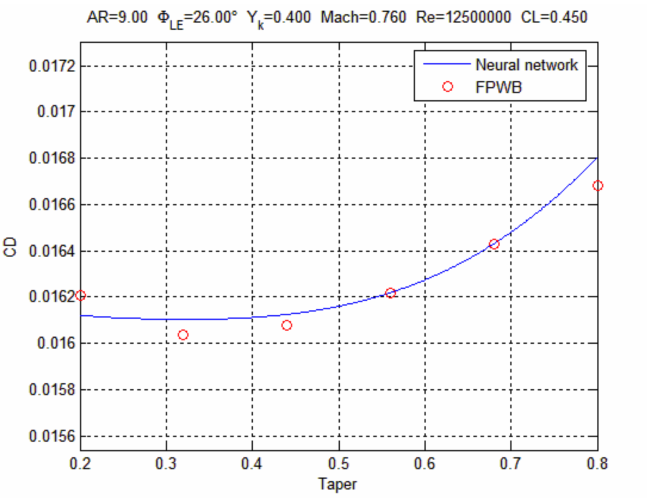

Fig. 24 - C_D variation on λ.

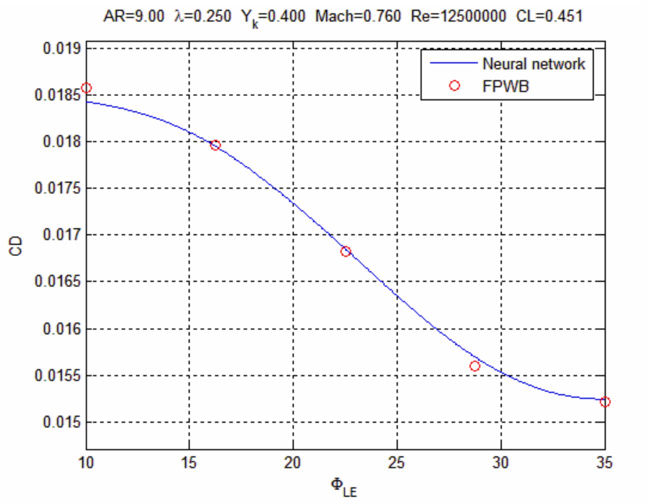

Fig. 25 - C_D variation on ϕ.

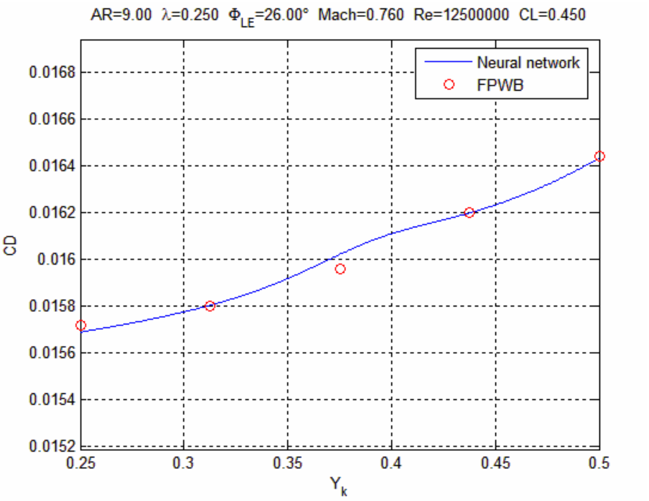

Fig. 26 - C_D variation on Y_K.

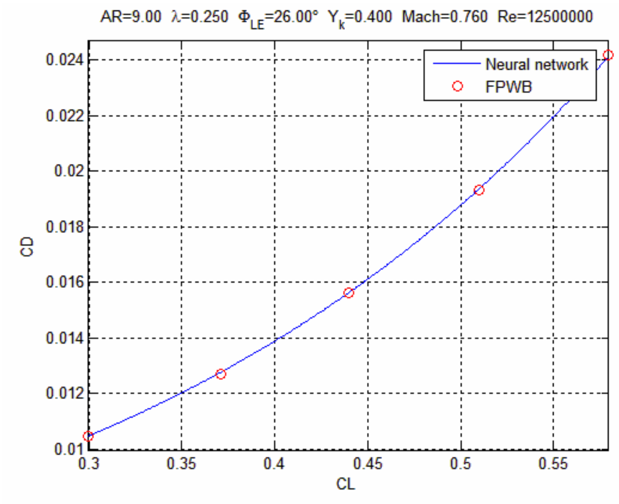

Fig. 27 - Drag polar for a typical wing-body configuration.

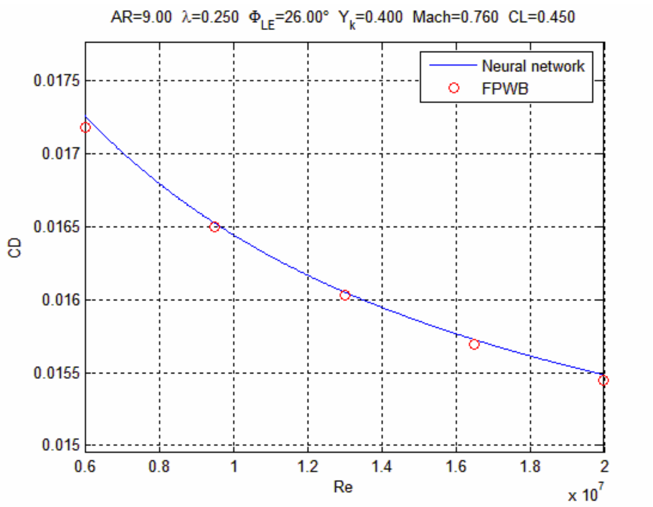

Fig. 28 - C_D variation on Re.

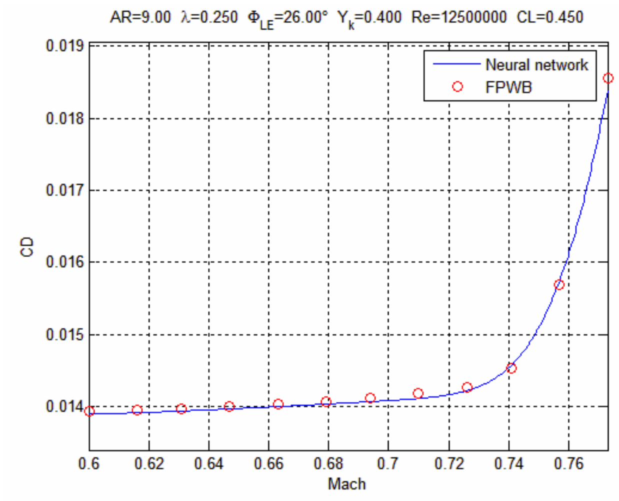

Fig. 29 - C_D variation on Mach. Drag divergence.

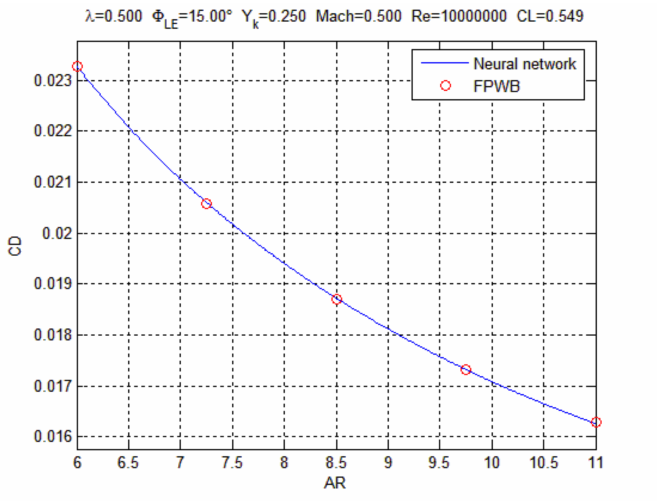

Fig. 30 - C_D variation on AR.

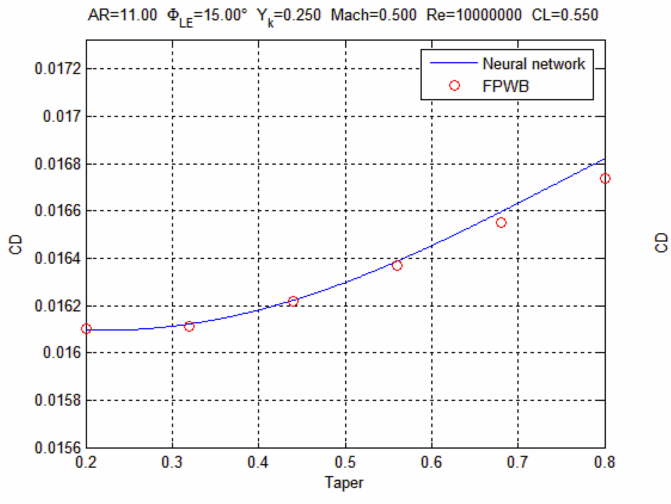

Fig. 31 - C_D variation on λ.

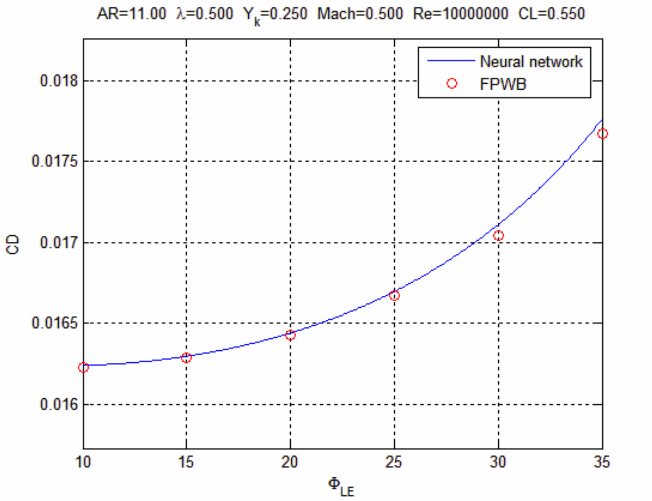

Fig. 32 - C_D variation on ϕ_{LE}.

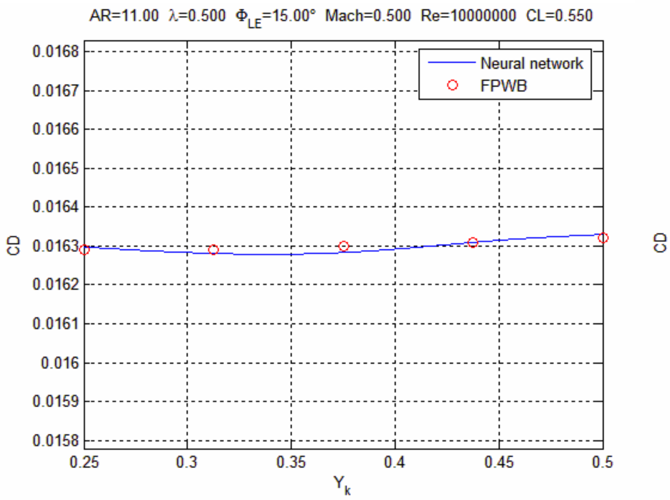

Fig. 33 - C_D variation on Y_K.

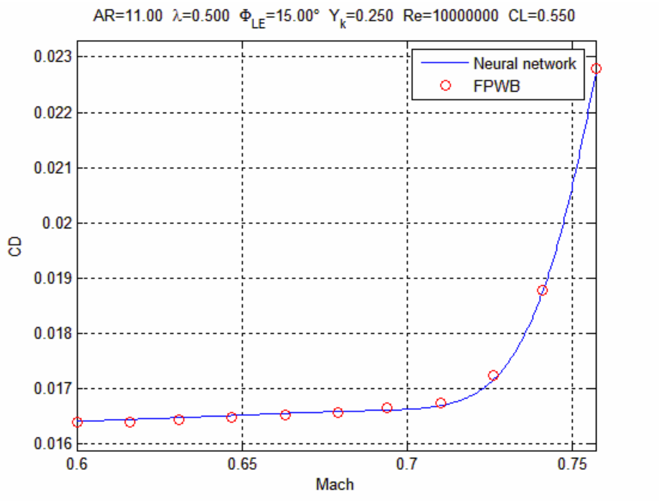

Fig. 34 - C_D variation on Mach number. Drag divergence.

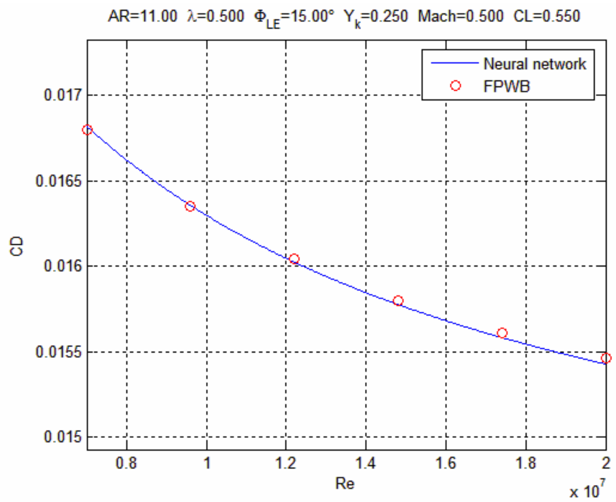

Figure 35. C_D variation on Re.

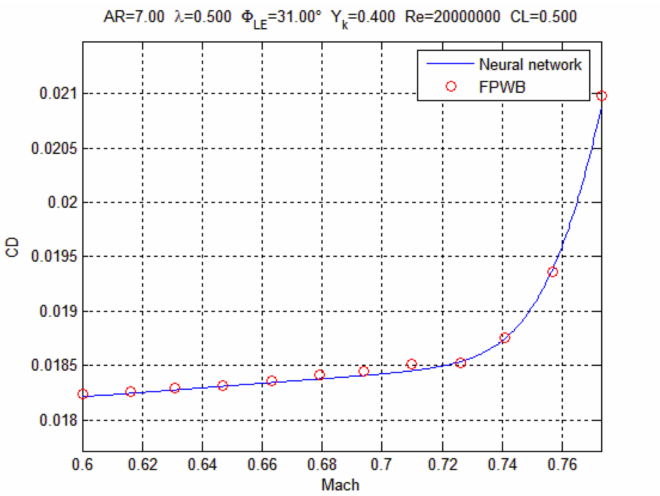

Fig. 36 - C_D variation on Mach number. Drag divergence.

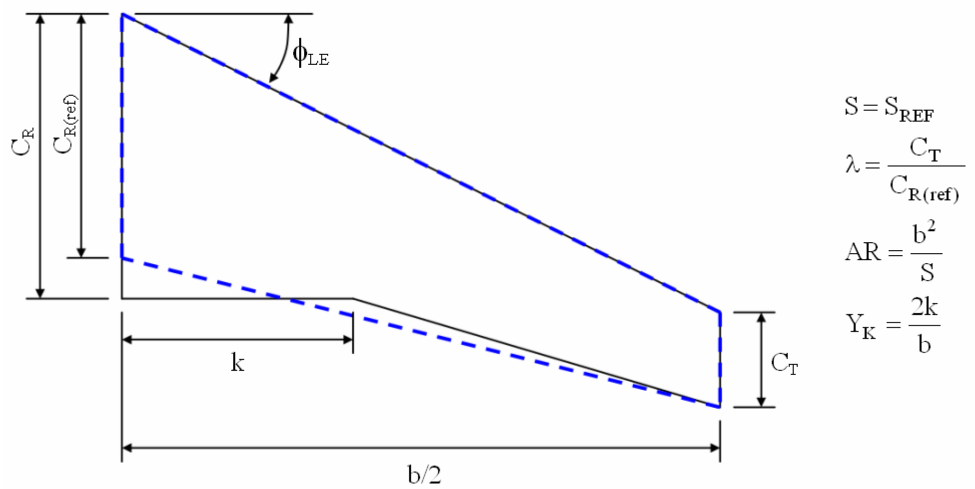

Figure 37 - Top right, ***the chosen parameters for the definition of the wing planform. Solid lines: actual wing; Dashed lines: reference wing (used for calculation of the taper ratio as input variable only).***

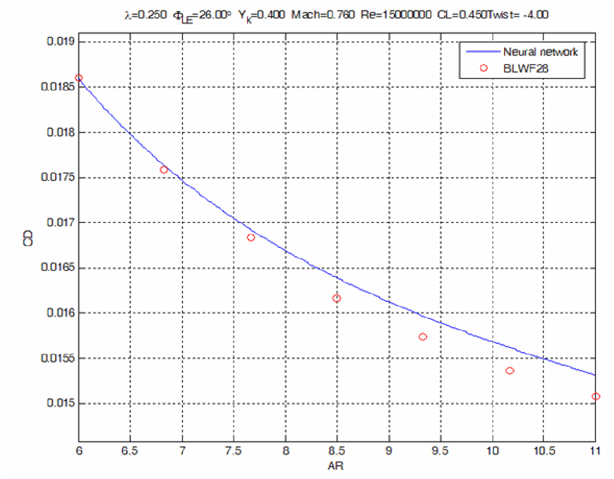

Fig. 38 – C_D prediction for wings of different aspect ratios. The twist angle for all wings is 4^o. Comparison is made with results of a full-potential code.

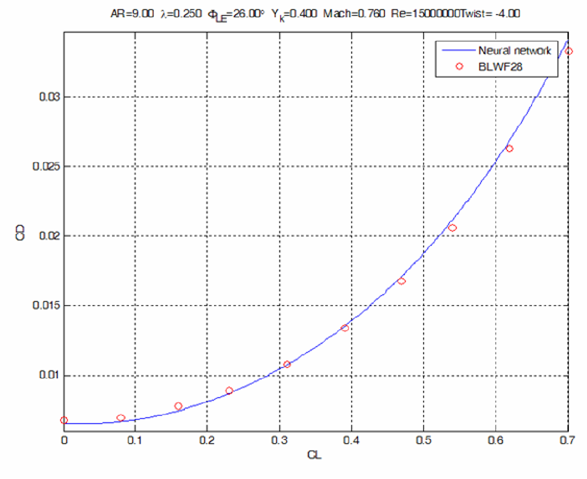

Fig. 39 – Predicted drag polar for a high-aspect ratio twisted wing. Comparison is made with results of a full-potential code.

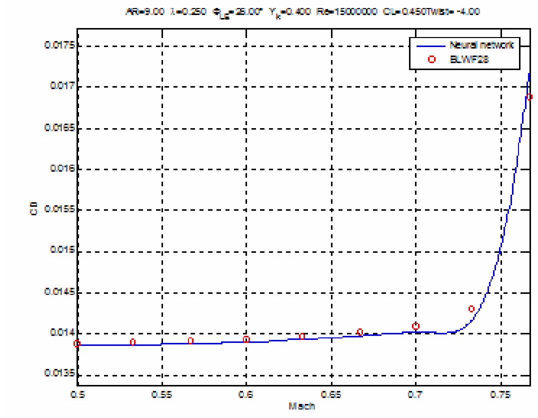

Fig. 40 – Drag divergence for a high aspect ratio twisted wing predicted by the MLP network. The twist angle for all wings is 4 degree.

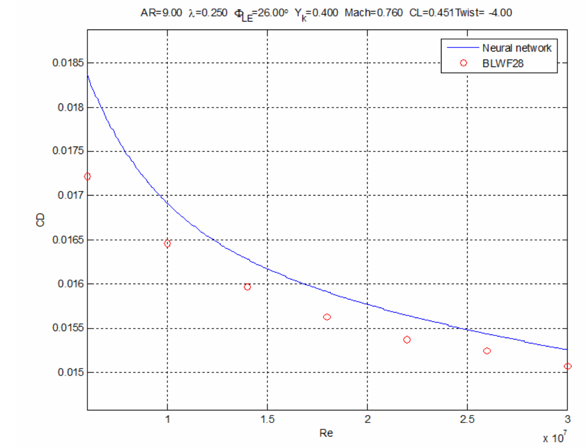

Fig. 41 – Predicted drag polar for a high-aspect ratio twisted wing.

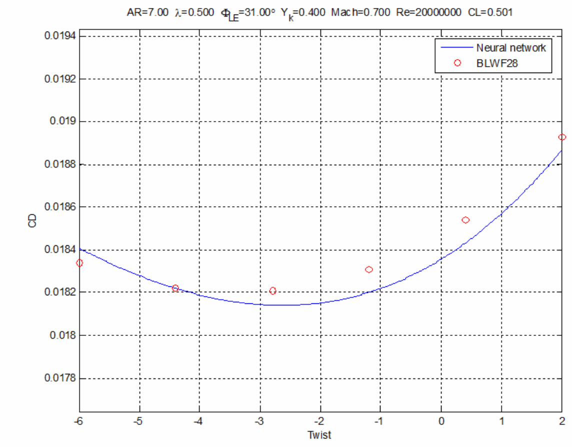

Fig. 42 - C_D variation with wing twist. AR = 7, Mach = 0.70, Re = $2\text{x}10^7$.

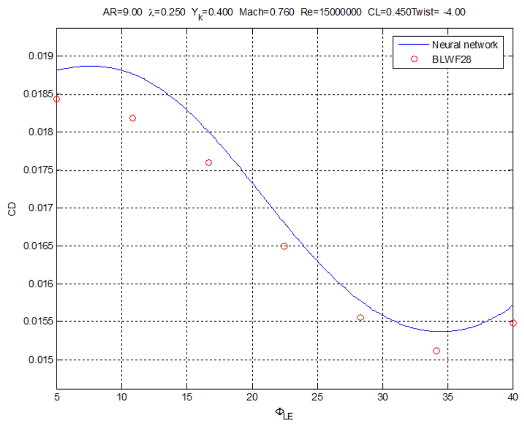

Fig. 43 - C_D variation with leading-edge sweepback angle. AR = 9, Mach = 0.70, Re = $2\text{x}10^7$.

7. Bibliography

1. Arbib, M.A., *The Handbook of Brain Theory and Neural Networks*. Second Edition ed. 2003, Cambridge Massachussets: The MIT Press. 1301.

2. Curvo, M. and Rios Neto, A. *Aerodynamic Modeling and Stochastic Adaptative Control of High Performance Aircraft Using Artificial Neural Network and Parameter Identification*. 2001, INPE - Instituto Nacional de Pesquisas Espaciais: São José dos Campos - Brazil.

3. Curvo, M. and Rios Neto, A. *Use of Functional Link Network in Aerodynamic Modeling of Aircraft*. in *16th Brazilian Congress of Mechanical Engineering*. 2001.

4. Isikveren, A.T., *Quasi-Analytical Modeling and Optimisation Techniques for Transport Aircraft Design*, in *Department of Aeronautics*. 2002, Royal Institute of Technology (KTH): Stockholm, Sweden. p. 354.

5. Kröse, B. and P.v.d. Smagt, *An Introduction to Neural Networks*. Eighth Edition ed. 1996, Amsterdam - Netherlands: The University of Amsterdam.

6. Roskam, J., *Airplane Design - Part VI*. Airplane Design. Vol. 6. 1985, Ottawa, Kansas: DAR Corporation.

7. Torenbeek, E., *Synthesis of Subsonic Airplane Design*. 1982, Rotherdam, Netherlands: The Delft Universty Press.

8. Giunta, A. A., Wojtkiewicz Jr., S. F. and Eldred, M. S., *Overview of Modern Design of Experiments Methods for Computational Simulations*, AIAA 2003-0649.

9. Drela, M., *XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils*, Conference on Low Reynolds Number Airfoil Aerodynamics, University of Notre Dame, June 1989.