ICAS-98-2,2,1

A98-31494

AN INVERSE DESIGN PROCEDURE FOR AIRFOILS USING ARTIFICIAL NEURAL NETWORKS

Neep Hazarika* Ismail H. Tuncer† David Lowe*

*Neural Computing Research Group, Aston University, Birmingham, B4 7ET, U.K.
†Department of Aeronautical Engineering, Middle East Technical University, 06531 Ankara, Turkey

ABSTRACT

In this paper, we investigate a novel method for the inverse design of airfoil sections using artificial neural networks (ANNs). Work on artificial neural networks has shown that ANNs can be used to emulate highly nonlinear relationships, such as that existing between surface pressures and the corresponding airfoil profiles in a flow. Surface pressure distributions generated by a panel code are used to train a neural network, which is then used to predict airfoil profiles for a given surface pressure distribution (the "inverse" problem), or to predict the pressure distribution for a given airfoil profile (the "forward" problem). The generalization capability of ANNs in the presence of noisy data is also studied. Results indicate that optimally trained artificial neural networks may accurately predict airfoil profiles and pressure distributions.

Introduction

In order to develop aircraft components and configurations which possess favorable aerodynamic performance characteristics efficient, automated design procedures are needed. At present, there are two main approaches to the design of aircraft configurations. The first is *direct optimization*, where an aerodynamic object function, such as the pressure distribution, is optimized computationally by gradually varying the design parameters, such as the surface geometry. The second is the so-called *inverse design methods*. Here, the pressure distribution at a given flow condition is specified, and the surface geometry is then determined to satisfy the pressure distribution.

In direct optimization methods, the flow solutions for various combinations of design parameters are determined by optimizing an aerodynamic object function, such as the pressure distribution. The computational effort can therefore be excessive. In order to keep the computational effort required within reasonable bounds, it is necessary to put limitations on the number of design variables. Details on the use of numerical optimization in aerodynamic design can be found in works by Hicks and other authors.[1–3]

In inverse design methods, the aim is to generate a geometry which corresponds to a prescribed flowfield information. Typically, a target pressure distribution is prescribed, and the body geometry which generates this pressure distribution is determined. There are many inverse techniques in use, for example, hodograph methods for two-dimensional flows,[4–7] and other two-dimensional formulations using panel methods.[8,9] In some inverse methods, the design of the body is carried out in an intermediate transformed plane.[10,11] The above methodologies have been extended to the three dimensional case.[12–15] Most analytical inverse design schemes are dependent on several external parameters that may reflect, for example, the various simplifying assumptions (such as linearization) made during the analysis phase. Further, it has been noted that if the inverse problem is ill-posed, small differences in the specified pressure distribution may lead to large differences in geometry.[16,17] Certain conditions, such as leading-edge stagnation conditions, may also have to be specified. Also, in certain cases, there may be no control over certain design parameters such as closure at the trailing edge of an airfoil profile.

The main objective of this work is to show that artificial neural networks can be trained to design airfoil profiles as well as to predict surface pressure distributions. Lately, neural networks have been used to predict aerodynamic forces and moments on an airfoil.[18,19] Since ANNs are trained to emulate underlying nonlinear processes, the problem of making simplifying assumptions or approximations hardly ever needs to be considered. ANNs have been shown to exhibit extremely robust behavior as far as generalization to new configurations is concerned. ANNs are also able to perform adequately even in the presence of noise in the data. These desirable characteristics of artificial neural networks formed part of the motivation for the exploration of ANNs as an inverse design tool. In the next section, we give a brief description of ANNs.

21st ICAS Congress
13-18 September 1998
Melbourne, Australia

Artificial Neural Networks

Artificial Neural Networks are computing systems based on a "learning" paradigm. They are also considered to be highly simplified models of the networks of real neurons in biological systems. An ANN consists of many individual "neurons," each of which processes an input information from other neurons and feeds its output into other neurons. A simple mathematical model for neural networks is given as:

$$n_i := g(\sum_j w_{ij} n_j - \mu_i)$$

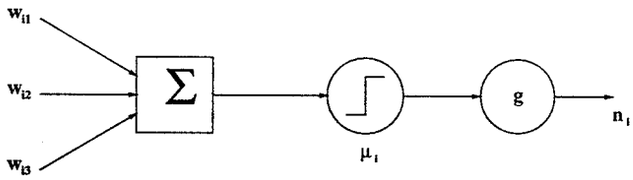

Figure 1. Schematic Diagram for a Simple Neuron

A schematic diagram for the model is shown in Figure 1. Here, n_i is called the state or activation of the neuron i and is continuous valued. $g(.)$ is a general nonlinear function called variously the activation-function, gain function, transfer function or squashing function. The weight w_{ij} represents the strength of the connection or synapse between neurons i and j. μ_i is the threshold value for neuron i, and is a cell-specific parameter. The general architecture of a two layer neural network with feed-forward connections and one hidden layer is shown in Figure 2. The input layer is not included in the layer count because its nodes do not correspond to neural elements. The weighted sum of the inputs must reach or exceed the threshold value for the neuron to fire or transmit.

A major issue in neural networks is the choice of appropriate connection weights so that the network can implement a specific task. This is known as learning. Many learning algorithms have been developed by which we can *teach* or *train* a network to perform a desired computation by iterative adjustment of w_{ij}. One drawback associated with neural networks is that it is normally very difficult to interpret the values of the connecting weights w_{ij} in terms of the task being implemented.

Neural networks offer a very powerful and general framework for representing nonlinear mappings from several input variables to several output variables. The form of the mapping is governed by a number of adjustable parameters (the weights w_{ij} described previously). The process of determining the values for these parameters on the basis of the available data is called *learning* or *training*, and the corresponding data set of examples is generally referred to as a *training set*. Since the central goal is to produce a system which makes good predictions for *new* data (referred to as a system which exhibits good *generalization*), we have to generate a second independent data set called a *test set*, which is produced in the same way as the training set, but with new values for, in our case, the angle of attack and the airfoil geometry. This is because of our basic assumption that the data on which we wish to use the trained neural network is produced by the same underlying mechanism as the training data.

Training generally involves minimization of an appropriate error function (*e.g.*, the root mean square (RMS) error) defined with respect to the training set. The aim of training is to determine the parameters of the neural network that has the best performance on new data. A simple approach to evaluate the performance of the network is to compare the error function on the independent test set. Since this procedure can itself lead to some over-fitting to the test set, the performance of the network should be confirmed by measuring its performance on a third independent set of data called a *validation set.*

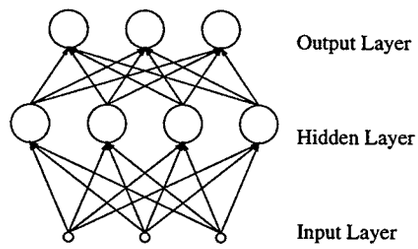

Figure 2. General architecture for a 2-layer Neural Network

For a successful implementation of artificial neural networks with reference to any particular problem, a number of issues have to be resolved; particularly important among these are:

- choice of learning algorithm;
- choice of network architecture;
- will the network generalize satisfactorily?

In general, we need to compute a set of weights w_{ij} that produce the desired outputs from each input pattern. Learning algorithms such as the back-propagation algorithm for feed-forward multilayer networks[20] help us to find such a set of weights by successive improvement from an arbitrary starting point. Thus, the choice of the learning algorithm depends mainly on the choice of the neural network for the particular task. Once the type of network and the learning algorithm is decided upon, we need to consider the network architecture. This is an important issue, because large networks tend to have poor generalization ability (the ability to *generalize* from examples on which it has been trained to examples previously unseen), but a network must be

large enough to have the expressive power necessary to accommodate the problem. Also, choice of appropriate representation of the input and output patterns can simplify the problem considerably.

The Inverse Design Procedure

An airfoil profile can be described by a set of x- and y-coordinates, as illustrated in figure 3. Given the airfoil profile, a pressure distribution may then be computed under various flow conditions set by the angle of attack α, freestream Mach number M_∞ and the Reynolds number R_e.

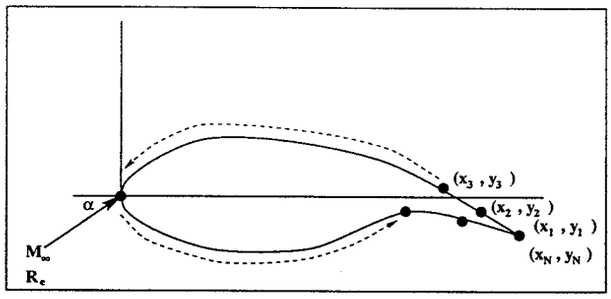

Figure 3. Flow field and airfoil data

In this work, we will only consider subsonic flows over NACA type airfoils at an incidence. Under normal circumstances, data from wind-tunnel tests would be used for training the network. In the absence of such explicit unclassified databases, however, we have simulated such data using a panel code, which computes the potential flow over an airfoil. Panel codes[21] assume the flow to be inviscid and incompressible, and they can adequately model subsonic flows at low angles of attack.

The ANN model used

In this study, a feed-forward neural network model was used to implement an inverse design procedure for airfoil profiles. Multi layered feed-forward networks have a set of input terminals (input layer) to feed the input patterns into the network. After the input layer, there can be one or more intermediate layers of units called hidden layers, followed by the final output layer where the computed results are obtained. Every unit feeds only the units in the next layer (feed-forward), and there are no back connections from the units in a layer to a previous layer. Figure 2 is an example of a 2-layer feed-forward neural network.

Once the type of network is chosen for a particular problem, we need to choose appropriate training regimes as well as an appropriate data representation and network architecture for the network to perform optimally.[20]

The ANN architecture used

Developing an ANN architecture appropriate to the task under consideration is a major concern, as each application requires its own architecture. With a good choice of network architecture, the trained network generates the right output for an unknown input. There are algorithms such as Fahlman and Lebiere's Cascade Correlation algorithm[20] which are capable of evolving a suitable architecture as part of the training procedure.

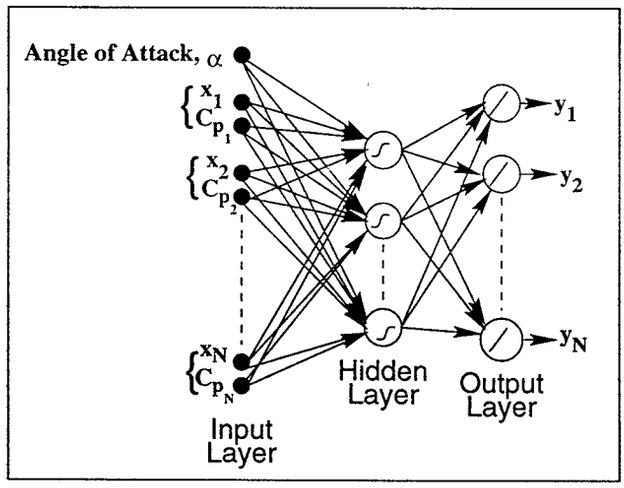

Figure 4. The Neural-Network Model trained to predict surface y-coordinates as a function of the airfoil chordwise pressure distribution and angle of attack α

The universal function approximation property of neural networks is employed to determine a nonlinear mapping that defines the relationship between the surface geometry, the external flow conditions and the surface pressure distribution. As shown in figure 4, the physical $x-$ coordinates of the airfoils, together with the orientation defined by the angle of attack α and the corresponding surface chordwise pressure distribution are used as inputs into a neural network, which is then trained to predict the corresponding surface y-coordinates.

During training, each input pattern is associated with the corresponding output pattern. Thus the network learns the actual functionality between the input stimuli and the output response, *i.e.,* it determines the relationship between the input and output patterns, in this case, the relationship between the pressure coefficients and the corresponding target upper-surface y-locations for each section.

Results and Discussion

In our investigation of neural network models for inverse design, we found that satisfactory results were obtained by using a feedforward, single hidden layer neu-

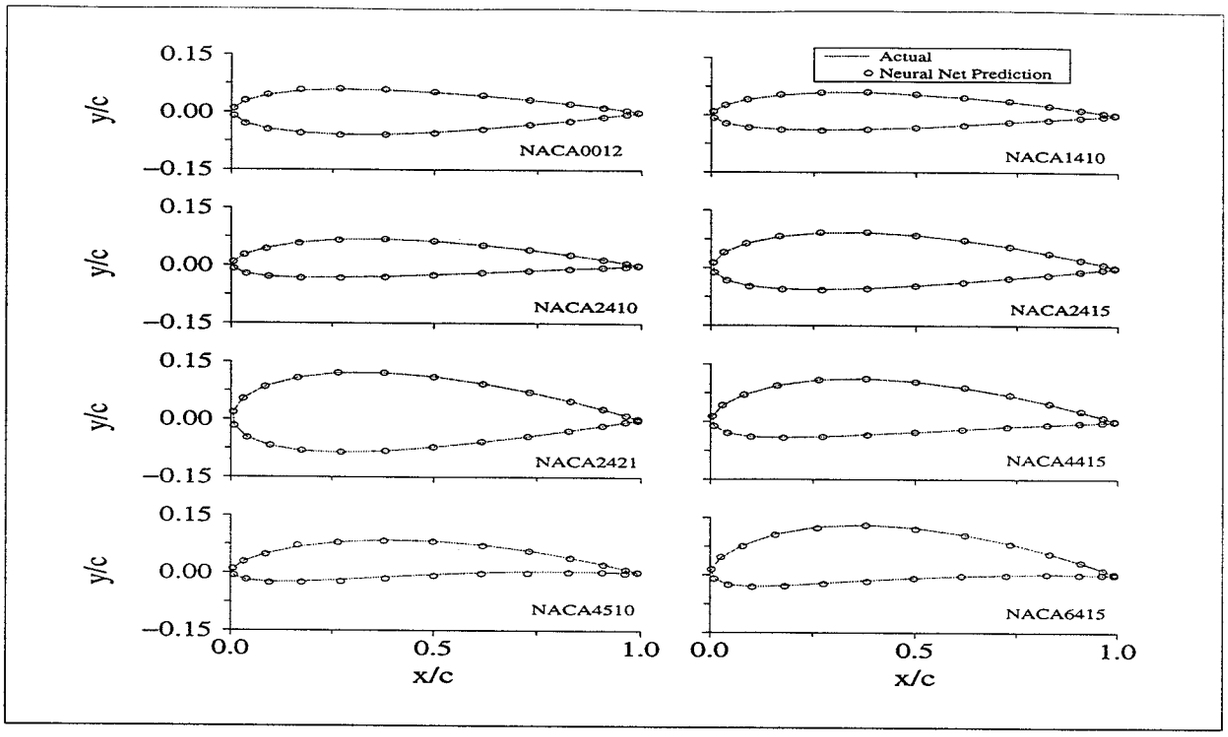

Figure 5. Comparison of predicted and theoretical airfoil profiles at zero degree angle of attack

ral network with a sigmoidal hidden activation functions, and linear outputs. Problems regarding the complexity of the model (determined in our case by the number of hidden units) using singular value decomposition of the hidden layer activations have been discussed elsewhere.[22,23] In our case, it was found that ten hidden nodes could adequately capture the nonlinear relationship between the airfoil profile, the external flowfield condition and the pressure distribution. The ANN was trained using a modification of the backprop algorithm[20] called "resilient propagation".[24] The use of an ANN, and the associated training algorithm, is typical in this type of prediction problem, and the details can be found in standard textbooks.[20] We will not describe these here.

As mentioned previously, training data are produced using a panel code, which computes the pressure coefficient C_p at desired locations on the airfoil surface. Thus, we have a database comprised of 26 upper- and lower-surface $x-$ and $y-$ coordinates, together with the simulated pressure distributions from 158 airfoil sections at angles of attack ranging from 0^o to 13^o degrees. There were 1422 patterns in total. The main goal is to determine the airfoil profile for a given pressure distribution under certain flow conditions, in this case, the angle of attack α. This is the "inverse" problem.

Figure 4 illustrates how the network is trained to learn the functionality between the surface geometry and flowfield information, and the corresponding surface pressure distribution. The network was trained to minimum error (using 1000 training patterns) on a test set (comprising 400 patterns) which was not used in the training process.

We show the results of applying the trained neural network on a separate validation set constructed from the remaining patterns. Eight such cases are shown in figure 5. Figure 6 shows how the training and testing root-mean-square (RMS) errors decrease with the number of minimization steps.

It is noticed that the present scheme is able to predict the airfoil profile as a function of the surface chordwise pressure distribution and the angle of attack α with a high degree of accuracy. A measure of the accuracy of the results obtained can be inferred from an examination of the maximum error, normalized as a percentage of the airfoil thickness, of the difference between the actual and the predicted profiles, defined as

$$\max_{i=1,\ldots,N} \frac{|y_i(\text{actual}) - y_i(\text{predicted})|}{\text{airfoil thickness ratio}} * 100\%,$$

where y_i(actual) is the actual y-coordinate of the section at location i, y_i(predicted) is the predicted y-coordinate, and N is the number of control points in the airfoil profile, in our case, 26. This is displayed in table 1 for the eight cases shown in figure 5.

Next, we solved the "forward" analysis problem, namely, prediction of the pressure distribution given a particular surface geometry as input. As before,

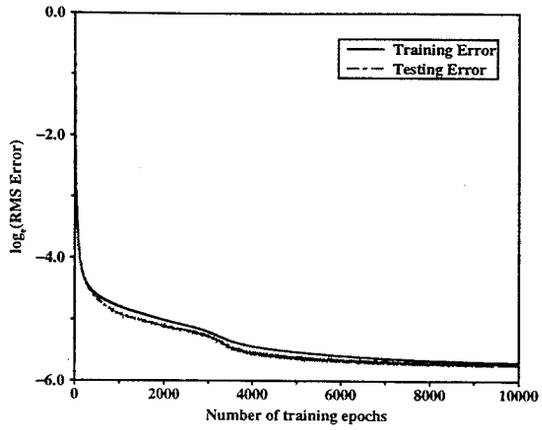

Figure 6. Convergence criteria for trained neural network. Note how the training and testing errors decrease with the number of minimization steps. The network was trained for 10000 epochs.

Airfoil	Maximum error (%)
NACA0012	0.407
NACA1410	1.087
NACA2410	0.8978
NACA2415	0.9476
NACA2421	0.5368
NACA4415	0.8808
NACA4510	4.836
NACA6415	2.047

Table 1. Maximum error normalized as a percentage of airfoil thickness for the validation cases shown in figure 5

we used the same 1000 patterns for training, and the same 400 patterns for testing. The remaining patterns were used for validation purposes. In this case, the network architecture is modified as shown in figure 7. Each input vector consists of the x- and y-coordinates of twenty-six locations on the airfoil surface, together with the angle of attack α, while each output vector consisted of the target chordwise pressure coefficients at the corresponding locations. The results are shown in figure 8. It is shown that reasonable accuracy can be achieved for all the validation cases.

In real-world situations, the data collection procedure involves implementing a series of wind-tunnel runs over various airfoils under different external flow conditions. The pressure distribution over the airfoil surface is measured by placing physical sensors in the form of probes, or advanced laser-Doppler anemometry techniques. The physical sensors then transduce the measured physical value to another quantity (in this case, the pressure coefficients at different locations on the airfoil), and digitize the results. A database is then

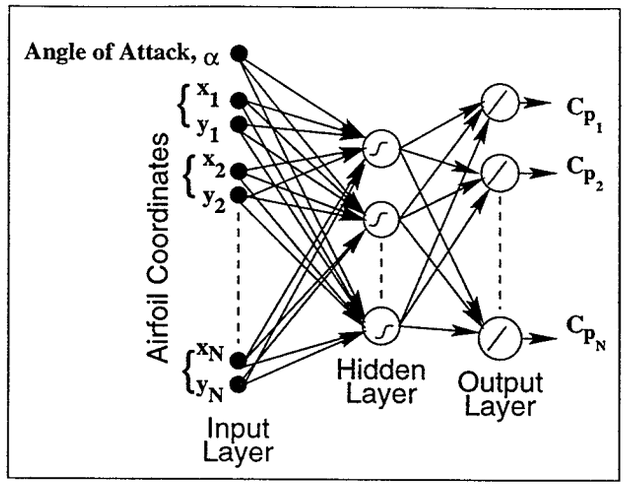

Figure 7. The Neural-Netwok Model trained to predict the pressure distribution as a function of the airfoil coordinates and the angle of attack α

built from these measurements. In the general case, our model has to perform an adequate inference of the airfoil's interaction with the flow field, and to do so within a changing operational context, even in the presence of noise related to the measured data.

An impressive feature of ANNs is that, since the network learns the underlying generator of the data, they can generalize even in the presence of noise, if care is taken to prevent overfitting. Therefore, it is possible to use experimentally measured pressure distributions as inputs to obtain very accurate solutions. In real-world data gathered from wind tunnel tests by sensors placed on the surface of the airfoils, instrumentation calibration and transduction can introduce a potential source of noise. This is simulated by adding a maximum of $\pm 20\%$ random noise to the values of the pressure distributions computed by the panel method. The goal is to determine whether the model developed can generalize in the presence of noise to the closest underlying distribution in an optimal sense.

Results, using the same training and testing criteria as before, are shown for the same eight validation cases in figure 9. Note the excellent generalization capability of the model.

This very important result indicates that it may be possible to utilize noisy data obtained from wind-tunnel experiments for design purposes, using pattern-recognition based algorithms such as artificial neural networks. In order to demonstrate the capability of neural networks as an inverse design tool under noisy data, the same simulated noisy data described previously is employed, in this case for the inverse design problem. Results are shown in figure 10. A comparison with figure 5 shows very little difference between the actual and predicted airfoil coordinate values, even

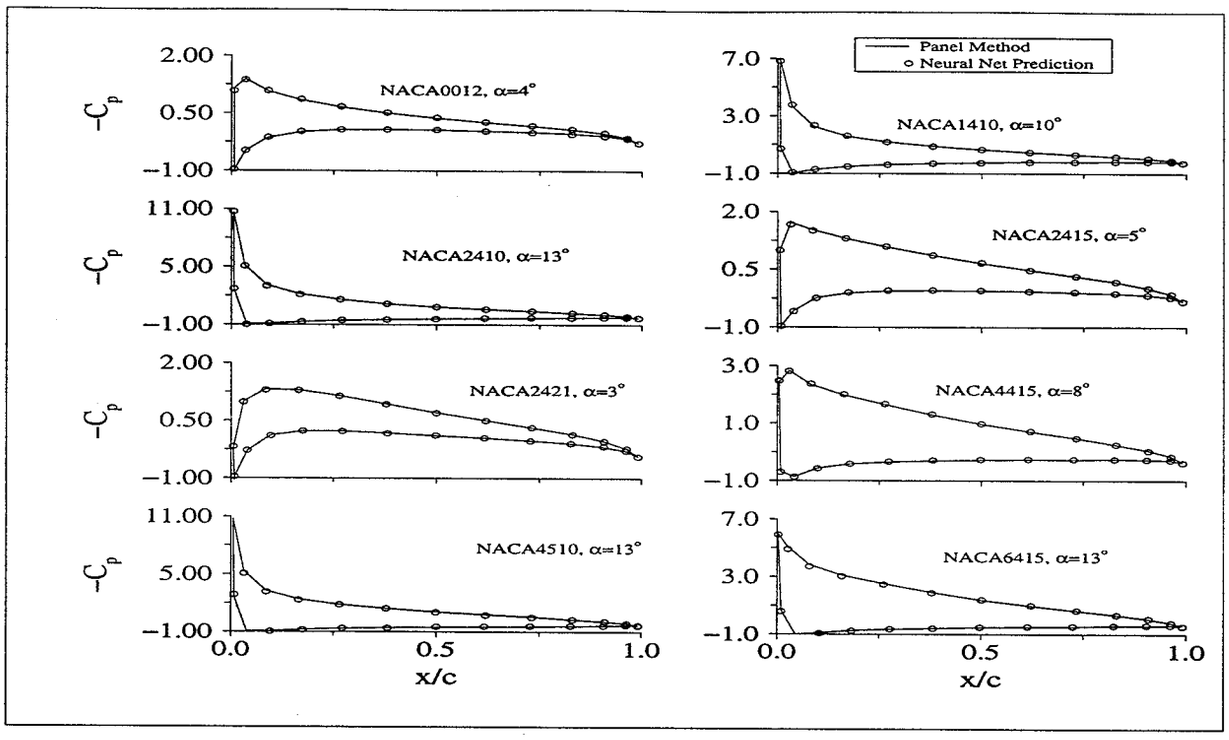

Figure 8. Theoretical and predicted pressure distributions using the present method

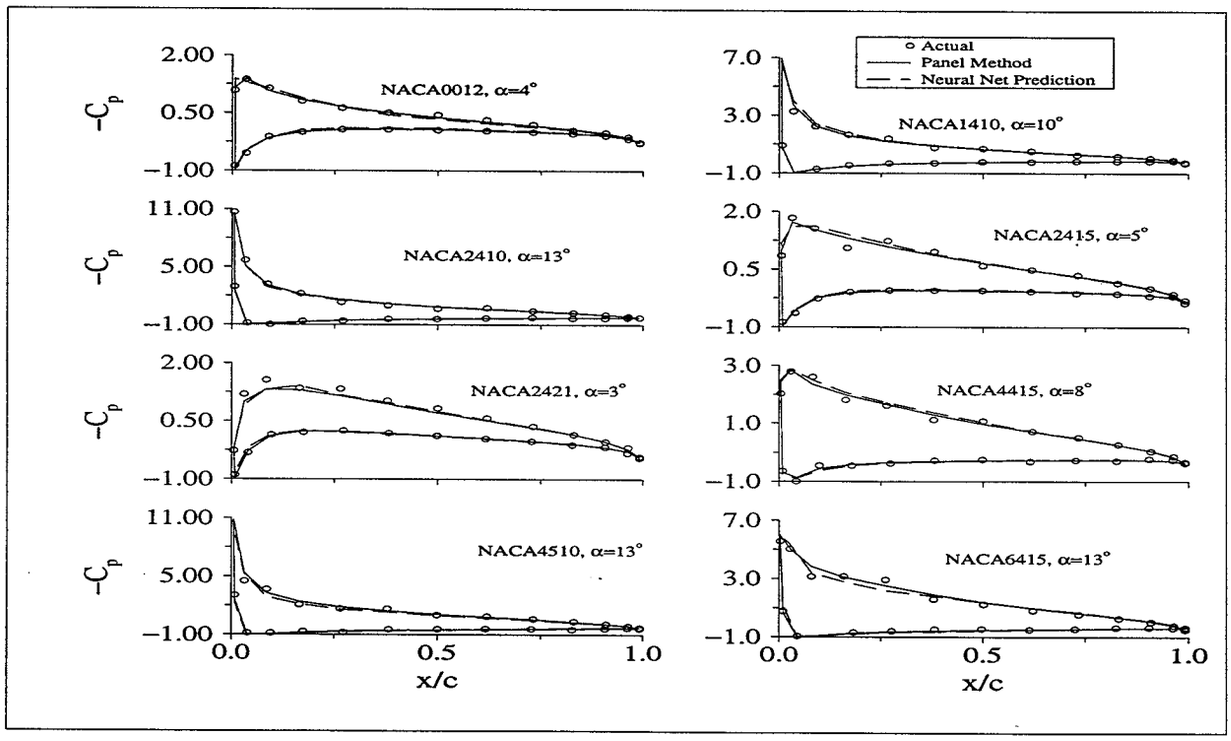

Figure 9. A comparison of the pressure distributions computed by the panel method and that predicted by the neural network for various flow conditions and airfoil contours using simulated noisy data

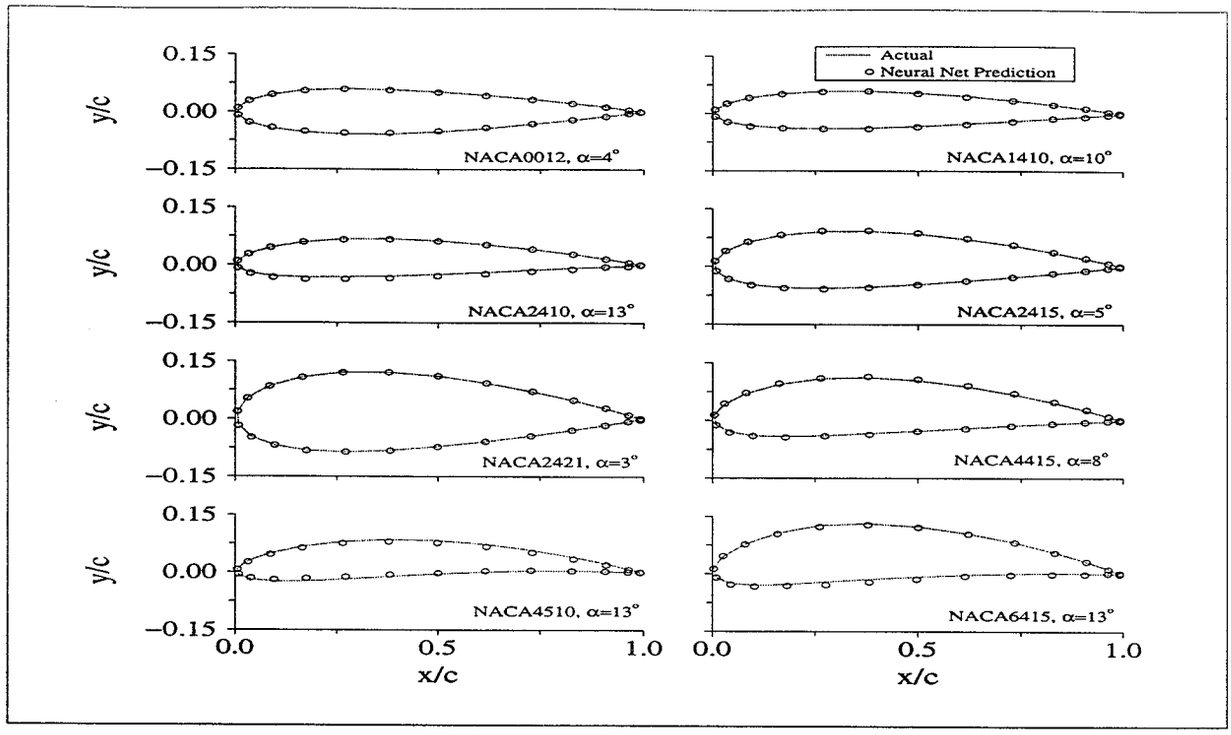

Figure 10. Actual and predicted sections using simulated noisy data

when noisy data is used.

Conclusions

In this paper, we have introduced an inverse design methodology using artificial neural networks which may be used for the design of airfoil profiles at subsonic flows. We used the data produced by a panel code to train the artificial neural network. The results indicate that the present technique can be used as a feasible inverse design tool, as well as an analysis tool.

Since the trained network can be used for inverse design in near real time, a potential application of this scheme is in the area of active or dynamic design under time-varying flow conditions. For instance, an ANN can be trained to predict the required airfoil geometry as a function of the flight conditions, so that a real-time adaption to changing flow conditions may be imposed. This can be particularly useful in the transonic regime, where the airfoil profiles can be dynamically deformed using *smart materials* to minimize or even preempt losses due to shock-boundary layer interaction. Further, the technique can also be extended to dynamically reduce aerodynamic interference between different aircraft components in time-varying flow conditions.

Acknowledgements

This work was supported under EPSRC contract K51815.

REFERENCES

1. Hicks, R.M. and Vanderplaats, G.N., "Design of Low-Speed Airfoils by Numerical Optimization," SAE Business Aircraft Meeting, Wichita, April 1975
2. Hicks, R.M. and Henne, P.A., "Wing Design by Numerical Optimization," AIAA Paper 77-1247, 1977
3. Vanderplaats, G.N., "CONMIN – A FORTRAN Program for Constrained Function Minimization," NASA TMX-62282, 1973
4. Nieuwland, G.Y., "Transonic Potential Flow around a Family of Quasi-Elliptical Airfoil Sections," National Lucht-en Ruimtevaart Laboratorium, Amsterdam, NLR-TR-T.172, 1967
5. Garabedian, P.R. and Korn, D.G.,"Numerical Design of Transonic Airfoils," *Numerical Solution of Partial Differential Equations – II*, Academic Press, 1971
6. Holst, T.L., Slooff, J.W. Yoshihara, H. and Ballhaus Jr., W.F., "Applied Computational Transonic Aerodynamics," AGARD-AG-266, 1982
7. Slooff, J.W., "Computational Procedures in Transonic Aerodynamic Design," Lecture presented at ITCS Short Course on Computational Methods in

Potential Aerodynamics, Amalfi,Italy, 1982 (also NLR MP 82020 U)

8. Ormsbee, A.I. and Chen, A.W., "Multielement Airfoils Optimized for Maximum Lift Coefficient," *AIAA Journal*, Volume 10, pp. 1620–1624, December 1972
9. Fray, J.M.J. and Slooff, J.W., "A Constrained Inverse Method for the Aerodynamic Design of Thick Wings with Given Pressure Distribution in Subsonic Flow," AGARD-CP-285, 1980
10. Tranen, T.L., "A Rapid Computer Aided Transonic Design Method," AIAA Paper 74-501, June 1974
11. Carlson, L.A., "Transonic Airfoil Design using Cartesian Coordinates," NASA-CR-2578, April 1976
12. Henne, P.A., "An Inverse Transonic Wing Design Method," AIAA Paper 80-0330, Pasadena, CA., January 1980
13. Shankar, V., Malmuth, N.D. and Cole, J.D., "Computational Transonic Design Procedure for Three-Dimensional Wings and Wing-Body Combinations," AIAA Paper 79-0344, January 1979
14. Garabedian, P., McFadden, G. and Bauer, B., "The NYU Inverse Swept Wing Code," NASA CR-3662, January 1983
15. Shankar, V., "A Full-Potential Inverse Method based on a Density Linearization Scheme for Wing Design," AIAA Paper 81-1234, 1981
16. Slooff, J.W. and Voogt, N., "Aerodynamic Design of Thick Supercritical Wings through the Concept of Equivalent Subsonic Pressure Distribution," NLR MP 78011 U, 1978
17. Slooff, J.W., "Wind Tunnel Tests and Aerodynamic Computations – Thoughts on their use in Aerodynamic Design," AGARD CP No. 210, Paper 11, 1976
18. Steck, James, E. and Rokhsaz, Kamran "Some Application of Artificial Neural Networks in Modelling of Nonlinear Aerodynamics and Flight Dynamics," AIAA Paper 97-0338, January 1997
19. Faller, William E., Smith, William E. and Huang, Thomas T., "Applied Dynamic System Modeling: Six Degree-of-Freedom Simulation of Forced Unsteady Maneuvers Using Recursive Neural Networks," AIAA Paper 97-0336, January 1997
20. Hertz, J., Krogh, A. and Palmer, R.G., *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Co., California, 1991
21. Hess, J.L., Johnson, F.T. and Rubbert, P.E., *Panel Methods*, Notebook, AIAA Professional Study Series, 1978
22. Lowe, D., "Characterising Complexity in a Radial Basis Function Network," *Proceedings of the IEE Fifth International Conference on Artificial Neural Networks* (Conference Publication No. 440), July 7-9, 1997, University of Cambridge, U.K., pp. 19-23.
23. Lowe, D. and Hazarika, N., "Complexity Modelling and Stability Characterisation for Long-Term Iterated Time Series Prediction," *Proceedings of the IEE Fifth International Conference on Artificial Neural Networks* (Conference Publication No. 440), July 7-9, 1997, University of Cambridge, U.K., pp. 53-58.
24. Riedmiller, M. and Braun, H., "A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm," Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, March 28–April 1, 1993