AIAA JOURNAL
Vol. 38, No. 1, January 2000

Aerodynamic Design Using Neural Networks

Man Mohan Rai* and Nateri K. Madavan†
NASA Ames Research Center, Moffett Field, California 94035

An aerodynamic design procedure that incorporates the advantages of both traditional response surface methodology and neural networks is described. The procedure employs a strategy called parameter-based partitioning of the design space and uses a sequence of response surfaces based on both neural networks and polynomial fits to traverse the design space in search of the optimal solution. This approach results in response surfaces that have both the power of neural networks and the economy of low-order polynomials (in terms of number of simulations needed and network training requirements). Such an approach can handle design problems with many more parameters than would be possible using neural networks alone. The design procedure has been applied to the redesign of a turbine airfoil from a modern jet engine. This redesign involved the use of 15 design variables. The results obtained are closer to the target design than those obtained using an earlier method with only three design variables. The capability of the method in transforming generic shapes, such as simple curved plates, into optimal airfoils is also demonstrated.

Introduction

ARTIFICIAL neural networks have been widely used in aeronautical engineering. Recent aerodynamic applications include, for example, flow control, estimation of aerodynamic coefficients, compact functional representations of aerodynamic data for rapid interpolation, grid generation, and aerodynamic design. Fan et al.[1] studied the possibility of achieving active laminar flow control using "smart walls," a combination of neural networks and microelectromechanical devices. Kawthar-Ali and Acharya[2] demonstrated the feasibility of using neural networks in the control of dynamic stall. Neural networks have also been used both to model unsteady flows and to optimize aerodynamic performance parameters.[3] The feasibility of reducing wind-tunnel test times by using neural nets to interpolate between measurements has been demonstrated by Norgaard et al.[4] This study showed that significant cost savings could be realized by using such an approach. A similar approach to data modeling and interpolation based on neural nets has been presented by Meade.[5] Neural networks have also been used to optimize the flap configuration for a multielement high-lift airfoil to achieve maximum lift.[6] Neural network applications in aeronautics are not limited to aerodynamics. A review of the applications of a variety of neural networks in structural analysis and design is provided by Hajela and Berke.[7]

A preliminary effort at designing airfoils using neural nets, where lift, drag, and pitching moment coefficients obtained from an inverse design code are used to train a feedforward neural network, has been reported by Huang et al.[8] The network is used to predict the variation of these aerodynamic quantities as a function of the angle of attack for both the cases used in training as well as those outside the training domain. No effort is made to obtain an optimal geometry for the given aerodynamic targets. In other work,[9] a neural network is used to determine, from a database of input pressure distributions, a pressure distribution that would produce the required flow conditions. An inverse design method is then used to compute the airfoil shape that corresponds to this desired pressure distribution.

Rai and Madavan[10] have explored the feasibility of applying neural networks to the design of turbomachinery airfoils. The focus of their effort was on developing a design process that utilized the ability of the computer to interpolate unstructured data in multiple dimensions—a task that humans find difficult to perform. Their process also permitted the designer to perform a variety of tradeoff studies rapidly. The principal idea behind their effort was to represent the design space, within some parameter limits, using a neural network, and then to employ an optimization procedure to search this space for a solution that exhibited optimal performance characteristics. This approach can be viewed as a variant of response surface methodology (RSM) where the response surface is constructed using a neural network. An excellent introduction to RSM can be found in the textbooks by Montgomery[11] and Myers and Montgomery.[12] Guinta et al.[13] provide an overview of the application of RSM to multidisciplinary optimization (MDO) and list several advantages of using RSM in MDO, namely, 1) the alleviation of problems associated with using noisy data in optimization studies, 2) the ease of integrating simulation tools from different disciplines, and 3) the efficient use of parallel computing resources. However, the authors[13] use polynomials to construct the response surfaces in their design efforts.

Although there are several methods that can be used to represent the functional behavior of design data, neural networks are particularly suitable for multidimensional interpolation where the data are not structured. Because most design problems in aerodynamics involve a multitude of parameters and datasets that often lack structure, neural nets provide a level of flexibility not attainable with other methods. In fact, partial datasets or even a single data point intermingled with more complete datasets can be used to influence the design process.

Aerodynamic design data have traditionally been obtained from a variety of sources. In the past, experiments and simple analyses have provided the majority of data used in design. More recently, the methods of computational fluid dynamics (CFD) have been used to generate a significant portion of the design data. A hierarchy of approximations to the governing partial differential equations, i.e., the Navier–Stokes equations, ranging from the simple potential flow equations to the Euler and Reynolds-averaged, Navier–Stokes equations, have been used for this purpose. Typically, the simpler and lower fidelity potential flow solutions have been used in the initial stages of design because they are relatively inexpensive to compute, and because a large number of solutions are required at this stage. Here the term *fidelity* is used to denote the extent to which the system of equations faithfully represents the physical characteristics of the flow. The higher fidelity Euler and Navier–Stokes solutions are generally used in the final stages of design because of the high cost of computing these solutions. A preliminary evaluation[10] of

Presented as Paper 98-4928 at the AIAA/USAF/NASA/ISSMO 7th Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, 2–4 September 1998; received 16 November 1998; revision received 21 May 1999; accepted for publication 21 May 1999.

*Senior Scientist, Information Sciences and Technology Directorate.
†Research Scientist, Numerical Aerospace Simulation Systems Division.

neural networks indicates that they provide a natural framework within which a succession of solutions of increasing fidelity can be represented and subsequently utilized for optimization and design, thus reducing overall design costs. The design data can come from a variety of sources, including experiments and computations. Rules-of-thumb that designers have evolved over a number of years can also be incorporated within the optimization routines as constraints. These facts are of considerable importance to the aircraft industry, which has accumulated enormous amounts of experimental data and numerous design rules over a number of decades.

This paper builds on the work reported by Rai and Madavan[10] where a simple two-layer feedforward neural network, a conjugate-gradient based optimization routine, and CFD data were used to design the midspan geometry of a turbine airfoil. The pressure distribution from a modern Pratt & Whitney (P&W) turbine was used as the desired target. The computed surface pressure distribution for the neural net-based airfoil design was found to be in good agreement with the target distribution.

For the design methodology of Rai and Madavan[10] to be used effectively, it is imperative that the objective function be populated both adequately and efficiently. A sparse population results in an inaccurate representation of the design space whereas an inefficient use of aerodynamic data in populating the design space could result in excessive simulation costs. The method[10] used to populate the design space was to choose an initial design point and then to perturb all of the design parameters about this point. This resulted in a geometric increase in the number of datasets required to represent adequately the design space as the number of design parameters increases. (In RSM nomenclature, this is a 3^m factorial design, where m is the number of design parameters.) Because the number of parameters required for aerodynamic design is relatively large, this approach is restricted to simple designs involving a few geometric parameters.

In this paper we explore the possibility of a parameter-based partitioning of the design space, where the functional dependence of the variables of interest, e.g., pressure, with respect to some of the design parameters is represented using neural networks, and the functional dependence with respect to the remaining parameters is represented using polynomials. For first- or second-order polynomials, the number of datasets required increases in a linear or quadratic manner, respectively, with the number of parameters. This procedure greatly reduces the number of datasets required to populate the design space and thus enables designs involving a larger number of parameters than would be possible using neural networks alone.

This paper also addresses issues related to the choice of the initial design point. Clearly, the further the optimal design is from this initial design point the larger the region of design space that needs to be represented by the response surface. An inordinately large number of data may then be required to populate the design space adequately. In this paper we use a sequence of response surfaces based on neural networks and polynomials that constitutes a search process. This approach is an extension of traditional response surface methods based on polynomials alone.[12] The number of simulations required in the current approach are relatively modest.

In our earlier work,[10] the design study was performed with three geometric variables only, thus resulting in some differences between the target and the design pressures. Here, we repeat the design study with 15 geometric parameters to provide much greater flexibility in defining the airfoil geometry. This translates into a design that is much closer to the target. A second study that demonstrates the ability of the method to transform a generic shape into the optimal airfoil is also presented. The design method as well as the results of these studies are described in this paper.

CFD Methodology

The ROTOR-2 computer program[14] is used to obtain the CFD simulations. The computational method used is a third-order-accurate, iterative-implicit, upwind-biased scheme that solves the time-dependent, Reynolds-averaged, thin-layer, Navier–Stokes equations. The flow region of interest is discretized using a system of patched and overlaid grids that exchange information during the solution process. The dependent variables are initialized to freestream values and the equations of motion are then integrated to convergence subject to the boundary conditions. The flow parameters that are specified are the pressure ratio across the turbine airfoil (ratio of exit static pressure to inlet total pressure), the temperature, unit Reynolds number and flow angle at the inlet, and the flow coefficient. Details regarding the solution methodology can be found elsewhere.[14]

Construction of Composite Response Surface

The primary motivation for constructing a composite response surface based on neural networks and polynomials comes from an examination of the relative strengths of these two approaches in interpolating design data. Neural networks provide a very general framework for estimation in multiple dimensions. Barron[15] provided the bound for the mean integrated squared error between a two-layer feedforward network (single hidden layer and linear output neuron) and a target function $f(x)$ as

$$O\left(\frac{C_f^2}{n}\right) + O\left[\frac{nd\,\ell n(N)}{N}\right] \tag{1}$$

where n is the number of neurons in the hidden layer, N is the number of training pairs, d is the number of dimensions over which the function is defined, and C_f is related to the Fourier transform of the function.

The mean integrated squared error is given by

$$\|f - f_{NN}\| = \int_{x_a}^{x_b} [f(x) - f_{NN}(x)]^2\,\mathrm{d}x \tag{2}$$

where $f_{NN}(x)$ is the input/output function corresponding to the trained neural network.

The first term in Eq. (1) is the approximation error and represents the difference between the target function and the closest (in terms of mean integrated squared error) neural network function with a given number of hidden layer neurons. The second term in Eq. (1) is the estimation error and represents the distance between this closest neural network and the neural network given by the weights estimated using the training data that are available. The approximation error can be reduced by increasing the number of hidden layer neurons. Because network training costs are typically a small fraction of the simulation costs, this error can be reduced sufficiently without unduly increasing the total computational cost. However, in neural-network-based design it is the second term that can have a large impact on the total computational cost. Because each simulation could require anywhere between a few minutes to several hours of computer time, reducing the estimation error by increasing N may not always be feasible.

Figure 1 shows the parabola given by

$$y = 2\left(x - \tfrac{1}{2}\right)^2 \tag{3}$$

and the neural network approximations to this function obtained with two neurons in the hidden layer and a single linear neuron for the output. The network was first trained with three training

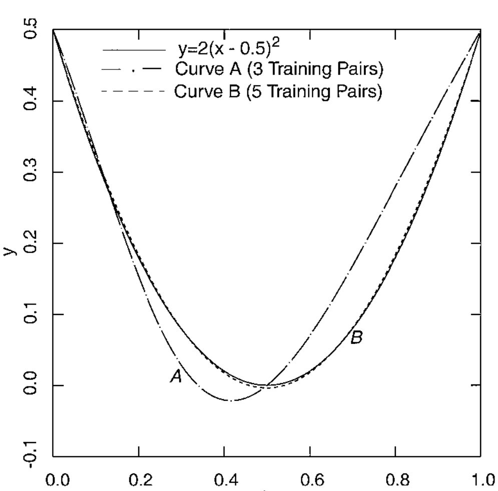


Fig. 1 Deterioration of neural network generalization capability with decreasing training data.

pairs (curve A) and then with five training pairs (curve B). The generalization ability obtained when only three training pairs are used is inadequate. It should be noted that the training error in this case (curve A) was decreased by 25 orders of magnitude. A marked improvement in generalization is seen with the use of five training pairs.

Figure 1 demonstrates that neural-network-based generalization can become unreliable when the amount of available training data is very small. However, the use of a single linear neuron with a preprocessor that provides the input nodes with the bias, the value of x and also x^2, would yield a perfect fit with just three training pairs. Note that such a single linear neuron is, in essence, a polynomial fit. The advantage of the polynomial fit provided by the single linear neuron is that it requires a prescribed minimum number of data points for a given number of polynomial terms, and this number increases in a polynomial fashion with the number of dimensions. For example, if a quadratic fit were used to represent the data, the number of data points required to compute the coefficients of the polynomial would increase quadratically with the number of dimensions. If the target function can be locally approximated using low-order polynomials, then there is an advantage to using polynomial fits instead of neural networks. Although Eq. (1) suggests that only a near-linear increase in the number of training data is required with an increasing number of dimensions, it is important to note that the number of hidden layer neurons n is typically large if the net is to represent the surface variation of aerodynamic quantities (21 hidden layer neurons were required in Ref. 10 to represent just three pressure distributions).

In the current approach, we combine conventional polynomial approximations on s-dimensional simplexes with the flexibility that neural nets provide. This results in a mathematical model whose complexity can be adjusted on a dimensional basis to suit the function being modeled, thus reducing the amount of data required. This approach assumes that the local variation of the design objective function with some of the geometric parameters can be accurately represented using low-order polynomials. The terminology s-dimensional simplex refers to a spatial configuration of s dimensions determined by $s + 1$ equispaced vertices, on a hypersphere of unit radius, in a space of dimension equal to s. By this definition, a two-dimensional simplex is an equilateral triangle that is circumscribed by a unit circle. The method of modeling functional behavior using polynomials whose coefficients are estimated from data defined on simplexes is referred to as a Koshal design.[12]

Consider a design scenario where the data can be generated for prescribed values of the design parameters. Additionally, assume that the variation of the aerodynamic data of interest with respect to some of the design parameters is not very complex (this may be because the parameter variations are small or because the underlying function is simple), and hence, does not require the generality of a neural network-based estimation scheme. In such a situation simple polynomials can be used to represent the variation of the function with these parameters and a neural network can be used for the remaining parameters.

The main thrust of this study is to develop this hybrid approach to represent aerodynamic data. We focus on two-dimensional airfoil design where the desired target is the surface pressure distribution. The three-layer neural network (with two hidden layers) shown in Fig. 2 is used for this purpose. The variation of aerodynamic data along the surface of the airfoil is typically far more complicated than the variation with small changes in geometric parameter values. Hence it is assumed in this design approach that the neural network will be used to represent aerodynamic data variation in physical space. The first node in the input layer is a bias node (input of 1.0). The second set of nodes is used to specify the physical location. In this particular two-dimensional design environment the physical location is specified by a single parameter, i.e., the axial location on the airfoil surface. Given t geometric parameters that determine the shape of the airfoil, assume that the variation of the first c parameters results in complex variations in the aerodynamic data and the variation of the remaining s parameters results in fairly simple variations in the data that can be represented by low-order polynomials ($t = s + c$). The third set of nodes in Fig. 2 accepts the first c geometric parameters. Pressure values corresponding to axial locations and geometry parameters specified at the input nodes are obtained at the output node.

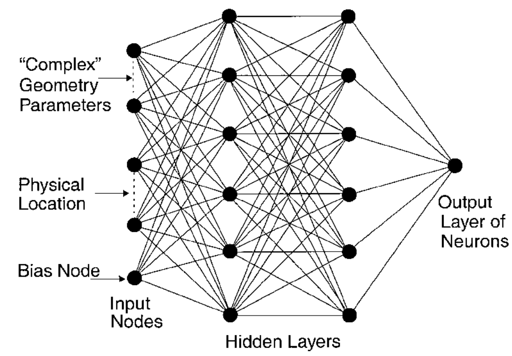
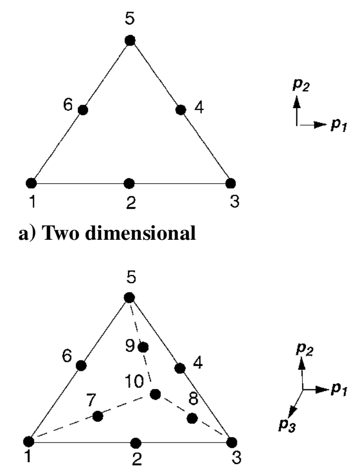


Fig. 2 Schematic of the three-layer feedforward neural network used in this study.



a) Two dimensional

b) Three dimensional

Fig. 3 Simplexes showing the points at which data are required for a quadratic approximation.

The variation of the aerodynamic data with the remaining s variables is approximated using simple polynomials. For a linear variation, the points at which the data are determined are located at the vertices of an s-dimensional simplex and are at unit distance from the origin. In this case there are $s + 1$ vertices and $s + 1$ unknown coefficients to be determined. For a quadratic variation, in addition to the vertices of the simplex, we include the midpoints of all of the edges as well. This results in $(s + 1)(s + 2)/2$ nodes and as many unknown polynomial coefficients that must be determined. Figure 3 shows the points at which these data are required in two dimensions ($s = 2$) and three dimensions ($s = 3$) for a quadratic fit.

In the two-dimensional case shown in Fig. 3 the pressure can be approximated as

$$p = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 \tag{4}$$

Given the pressure values $p_1, p_2, \ldots, p_6$ at the vertices of the simplex, the coefficients $a_1, a_2, \ldots, a_6$ can be obtained from the following system of equations:

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_6 & y_6 & x_6^2 & x_6 y_6 & y_6^2 \end{bmatrix} \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_6 \end{bmatrix} = \begin{bmatrix} p_1 \\ \cdot \\ \cdot \\ \cdot \\ p_6 \end{bmatrix} \tag{5}$$

The generalization to higher-dimensional simplexes and higher-order polynomials is straightforward. Clearly, a certain minimum number of pressure values is required depending on the number of dimensions and the order of the polynomial used. However, the number of pressure values that are typically used is more than this minimum (greater than six in the preceding example). This helps reduce the effect of noise in the data on the accuracy with which the response surface models the functional relationship between the aerodynamic variable, e.g., pressure, and the geometric parameters. The coefficients in this case are estimated using a least-squares approach. Guinta et al.[13] recommend using 1.5–2.5 times the minimum number of pressure values.

The method of combining neural networks and traditional polynomial fitting techniques is given next:

1) Obtain simulation data at each of the vertices of the simplex used for the polynomial fit. Multiple simulations will be required at each vertex if some of the geometric parameters are represented by the neural net.

2) Assign one neural net for every vertex of the simplex. Train each neural net with the simulation data generated for the corresponding vertex. The input for each net includes the bias value, the axial location along the airfoil, and the "complex" geometric parameters.

The following procedure is used to obtain the pressure corresponding to a given axial location and a set of geometric parameter values:

1) Obtain the pressure at each of the vertices of the simplex using the corresponding neural nets. This is easily done because the axial location on the airfoil and the values assigned to the complex geometric parameters are known.

2) Compute the estimates of the polynomial coefficients, e.g., as in Eq. (5), and then use the prescribed values of the simple geometric parameters to obtain the estimate for the required pressure value, e.g., as in Eq. (4).

Because the error that is being minimized during the design process is simply a function of the target pressure values and the estimated pressure values on the surface of the airfoil, this error can be computed by repeating steps 1 and 2 at each of the surface points at which the target pressure is prescribed.

The trained networks together with the polynomial fit constitute the composite response surface. The accuracy with which this response surface represents the actual functional dependence of the aerodynamic quantities on the design parameters is determined by the accuracy of the CFD simulations, the number of simulations used to populate the design space, the network parameters such as the number of neurons in the hidden layers (Fig. 2), and the order of the polynomial used. The accuracy of the CFD simulations will not be discussed here. The accuracy with which the networks represent the training data is given by the training error *TE*, which is minimized to obtain the network weights. For any one of the neural networks, this training error is given by

$$TE = \sum_{n=1}^{nmax} \sum_{i=1}^{imax} \left(\tilde{p}_i^n - p_i^n\right)^2 \tag{6}$$

where $\tilde{p}_i^n$ is the set of target pressures, p_i^n is the output pressure from the network, *imax* is the total number of points on the surface of the airfoil at which the target pressures are prescribed, and *nmax* is the number of CFD solutions used to train this particular network. Note that *nmax* has to be large enough that the functional dependence of the pressure on the complex variables is modeled accurately by the neural networks. The number of neurons is increased successively until the training error is sufficiently small. The training error was reduced by about four orders of magnitude for each of the design studies. This reduction in the training error was deemed adequate (results are presented in the next section).

The accuracy with which the composite response surface represents the design space can be estimated as follows:

1) Compute validation datasets (different from the training set).

2) Obtain the pressure distributions for these validation cases using the response surface.

3) Compare the pressure distributions obtained in steps 1 and 2.

This validation process is essential in establishing the adequacy of the generalization capabilities of the composite response surface. Typically, this is carried out at the centroid of the simplex.

Current Design Method

The current aerodynamic design strategy (for a prescribed target pressure distribution) can be summarized as follows:

1) *Define the target pressure distribution.* This is obtained from a set of higher-level design requirements.

2) *Determine an initial geometry with a pressure distribution that is close to the target.* There are several approaches, such as inverse design procedures,[16,17] meanline analyses, and streamline curvature-based methods,[18] that can be used to obtain this initial geometry. This geometry serves as the centroid of the first simplex.

3) *Populate the design space in the vicinity of the initial geometry.* Construct a simplex in design space around this centroid and obtain CFD solutions at each of the vertices (for a linear variation) and interior nodes (for quadratic or higher-degree polynomials).

4) *Train the neural networks and compute the polynomial coefficients.* The input nodes of the neural nets will typically contain parameters that correspond to the physical location on the airfoil and those geometric parameters that give rise to complex surface pressure variations. Train the neural nets and compute the polynomial coefficients that define the pressure variation within the simplex.

5) *Search the region of the design space represented by the composite response surface.* Various methods can be used to accomplish this constrained search. A conjugate gradient method was used in this study. Geometrical and other constraints and rules of thumb that designers have evolved can be incorporated within this search procedure. As an example, a penalty function method was used to enforce an inlet angle constraint in Ref. 10. In addition, constraints that limit the search procedure to the volume of the simplex are also incorporated in the search. Note that, on occasion, it may be advantageous to permit the search to extend outside the domain of the simplex. If this extended search culminates outside the simplex, a CFD simulation must be performed to determine the value of the objective function at this local optimum to confirm that it is indeed lower than anywhere inside the simplex. This confirmation is required because a gradual degradation of the generalization capability of the response surface can be expected as one moves outward from the boundaries of the simplex.

6) *Relocate the simplex.* If the local optimum obtained in the preceding step lies on or outside the boundaries of the simplex, then this point is chosen as the new centroid and steps 3–6 are repeated until the search culminates inside the simplex. However, the process can be stopped at any time when the design is deemed adequate.

7) *Refine the solution.* Several types of design refinements may be necessary. A particular design may require many iterations of steps 3–6 before the optimal solution is obtained. One reason for this could be that the initial design is very different from the target. The need to minimize overall design costs dictates that one obtain a preliminary design based on low-fidelity, low-cost simulations, e.g., potential flow solutions. Steps 3–6 are then repeated using higher fidelity simulations, e.g., Euler or Navier–Stokes solutions. A second level of refinement may involve repeating steps 3–6 with a simplex of reduced size.

8) *Validate the design.* As a final step in the process the geometry corresponding to the optimal design is used to obtain the pressure distribution (via a CFD simulation). This computed pressure distribution is compared with the target pressure distribution to determine the adequacy and quality of the design.

9) *Perform design tradeoff studies, as required.* The optimal design obtained in steps 1–8 is a point in design space that meets all of the initial design criteria. However, after obtaining this optimal design, the designer often wishes to modify the target or the constraints to arrive at a better and improved design, or analyze a variety of what-if scenarios. Several hundred such tradeoff studies may be required before the final design is defined. These analyses can be performed very efficiently by representing the functional dependence of the aerodynamic quantities in the vicinity of the design obtained in steps 1–8 using the neural net/polynomial approximation and once again searching this space with the new targets and constraints embedded in the search procedure. Clearly, this process can only be used if the new targets are contained in the region of design space where the generalization capabilities of the response surface are adequate. Our experience with aerodynamic design has been that the search procedure requires two to three orders of magnitude less computing time than that required for simulation and training the network. This allows the designer to perform a variety of tradeoff studies rapidly that would naturally involve

changing the constraints to resolve design conflicts or improve the design.

The preceding methodology combines neural nets and polynomial fits to represent the variation of the objective function in design space and low- and high-fidelity simulation techniques to reduce overall computing costs. These combinations result in a powerful design methodology that could make large-scale aerodynamic design tractable.

Comparison of Current and Earlier Design Methods

In Ref. 10, the authors laid out the framework of a neural net-based aerodynamic design capability that comprised the following elements: 1) Identify an initial design point, 2) populate the design space using simulations and /or experiments, 3) train the neural network, and 4) find the optimal solution by searching the region of the design space in which the functional dependence of the aerodynamic quantities is modeled by the network. Preliminary procedures for implementing these steps were also presented. The current design methodology employs techniques that vastly improve the implementation of steps 1–3.

The use of a sequence of response surfaces that constitutes a search process allows the designer to pick an initial design point that is not close to the optimal design, and subsequently traverse the design space to get close to this optimal point. As a result the choice of the initial design point is far less critical in the current method than in that of Rai and Madavan.[10]

In Ref. 10 the design effort used 27 pressure distributions corresponding to 27 airfoil geometries to populate a three-dimensional design space. The airfoil geometries were obtained by using three values (obtained by perturbations about a mean) for each of the three geometric parameters. Clearly, this process results in inordinately large number of simulations in high-dimensional design spaces. For example, the number of simulations that would be required for a 100-parameter design problem is 3^{100}, or about 5×10^{47}. On the other hand, for a linear representation within a simplex, the current approach would require a few hundred simulations.

In addition to drastically reducing the computational requirements to obtain the CFD simulations, the current approach also has a dramatic impact on the neural net training process. First, the reduction in the total amount of simulation data greatly reduces the training requirements. Second, the use of one neural net to represent each pressure distribution (as opposed to a single neural net that represents all of the pressure distributions as a function of axial distance and geometry parameters) also contributes to reduced training times. This is because a part of the complexity of representing the function (in this case, surface pressure variation with geometry parameters) is transferred from the neural net to the polynomial approximation. Figure 4 illustrates this latter point by comparing training times for

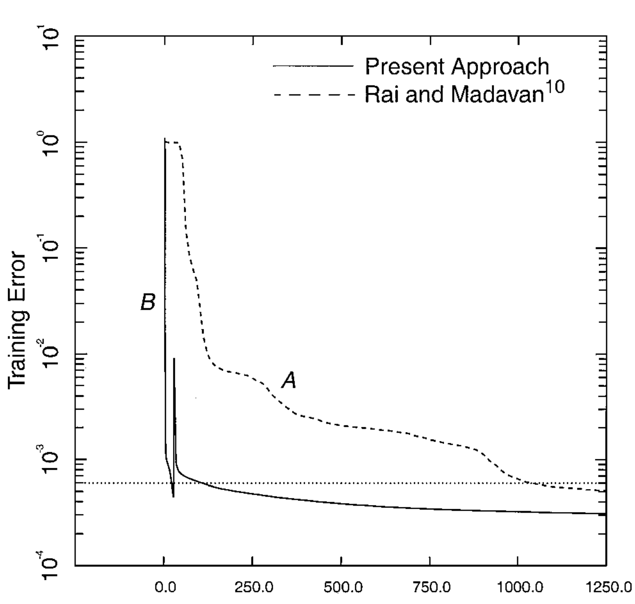


Fig. 4 Comparison of neural net training times required for the present approach and that of Rai and Madavan.[10]

the 27 simulations of Rai and Madavan.[10] In the first training process, all 27 pressure distributions were represented by a single net (curve A). Twenty-seven individual nets were used in the second training process. Curve B represents the variation of the sum of the 27 corresponding training errors with cumulative training time. A comparison of curves A and B shows about an order of magnitude reduction in training time. Note that the computational time required to implement the polynomial approximation is negligible.

In Fig. 4 curve B shows a sudden increase in the value of the training error at about $t = 50$ CPU s. This is because the initial values for the connection weights of the 27 networks is obtained as follows: 1) A single network with the same network connections as the other nets is first trained using pressure data obtained for the centroid of the simplex (cube, in this case). 2) The connection weights from this network are then used to initialize all of the nets. This approach to weight initialization further reduces the total training time requirements. The sudden increase in the value of the training error represents the point in time at which the weights of the single trained net are transferred to all of the other nets.

Parametric Representation of Airfoil Geometries

Because this study focuses on airfoil design, the ability to represent various airfoil geometries with a common set of geometrical parameters is essential. These parameters must be chosen such that variations of the airfoil geometry can be obtained by smoothly varying them. Geometrical constraints imposed for structural reasons, or aerodynamic reasons, e.g., to eliminate flow separation, should be included in this parametric representation as much as possible. Additionally, the smallest number of parameters should be used to represent the family of airfoils.

In Ref. 10 the airfoil geometry was represented using the thickness distribution and the mean camber line. The airfoil thickness is defined at specified control points and thickness values at other points are obtained using a tension spline. Different airfoil shapes can be obtained by varying the thickness values at the control points. Similarly, control points on the mean camber line are defined and a tension spline is used to obtain other points on the camber line. Again, different airfoil shapes are obtained by varying the location of these control points. The tension in the tension splines in both cases is adjusted to prevent undesirable airfoil shapes. The tension values can also be considered as additional geometric parameters, but this was not done in Ref. 10. This method of airfoil generation is referred to as scheme A later in the text.

In this study, we use the preceding approach to compare results obtained using the current design methodology with those of Rai and Madavan.[10] However, this method of airfoil generation was found to be inflexible and prone to generating inadmissible airfoil geometries when the initial geometry was far from optimal. A second, more robust, method of airfoil generation was developed to solve this problem. This new approach is illustrated in Fig. 5 and has the following salient features:

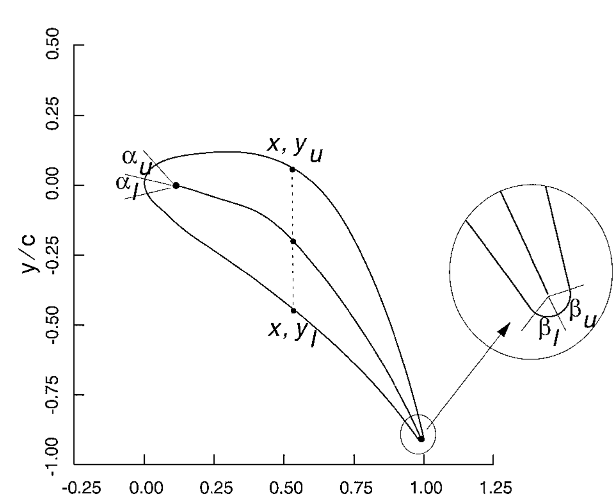


Fig. 5 Typical airfoil geometry showing location of nodal points on the airfoil surface and the defining angles in the new method of geometry parameterization (scheme B).

1) The leading edge is constructed using two different ellipses, one for the upper surface and one for the lower surface. The eccentricity of the upper ellipse and the semiminor axes of both ellipses are specified as geometric parameters (e_u, t_u, and t_l), respectively. All other related parameters can be determined analytically. The major axes of both ellipses are aligned with the tangent to the camber line at the leading edge. This tangent is initially aligned with the inlet flow but is allowed to rotate as the design proceeds. The angles α_u and α_l determine the extent of the region in which the leading edge is determined by these ellipses. The two ellipses meet in a slope-continuous manner.

2) The trailing edge can also be constructed in a similar manner with the major axes of the ellipses aligned with the tangent to the camber line at the trailing edge. However, in this study the trailing edge was defined using a single circle. The angles β_u and β_l determine the extent of the region in which the trailing edge is determined by this circle.

3) The region of the upper surface between the upper leading edge ellipse and the trailing edge circle is defined using a tension spline. This tension spline meets the leading edge ellipse and the trailing edge circle in a slope-continuous manner. Additional control points for the tension spline that are equispaced in the axial direction are introduced as necessary. These points provide additional control over the shape of the upper surface. The lower surface of the airfoil between the lower leading edge ellipse and the trailing edge circle is obtained in a similar manner.

This new method of generating the airfoil (scheme B) was found to be robust and provided a smooth transition from a constant thickness curved plate to the optimal P&W airfoil.

Turbine Airfoil Design Using Earlier and Current Methods

Earlier Design

The turbine airfoil design study presented in Ref. 10 is briefly reviewed here first. The goal of this study was to design an airfoil that would yield a prescribed target pressure distribution.

The target pressure distribution was supplied by P&W (private communication, F. Huber, 1997). This pressure distribution was obtained at the midspan of a turbine vane from a modern jet engine. The following flow and geometry parameters were also supplied by P&W and used in the current design process: inlet total pressure, inlet temperature, exit Mach number, inlet and exit gas angles, axial chord, leading and trailing edge thickness values, eccentricity of leading edge ellipse, radius of midspan section, and the number of vanes in the row.

Initiating the design process required the generation of several airfoils whose pressure distributions formed an envelope around the target pressure distribution. To do this an initial design that was relatively close to the target was obtained using a trial-and-error procedure. Three control points were chosen corresponding to the axial location where the airfoil thickness is maximum, the midpoint between the maximum thickness point and the start of the camber line, and the midpoint between the maximum thickness point and the end of the camber line. The thickness values at the ends of the camber line were provided by P&W and specified a priori. The thickness values at the control points were varied to obtain the training set (three values of each). Thickness values at other points on the camber line were obtained using a tension spline. A total of 27 airfoil geometries and corresponding pressure distributions were computed.

Figure 6 shows the envelope of airfoil pressure distributions obtained from CFD simulations for this set of airfoils and the target pressure distribution. The neural net was trained using these 27 pressure distributions and then used to predict the optimal design. Figure 6 also shows that the pressure distribution obtained from the CFD simulation for this optimal design is in good agreement with the target distribution.

Current Design

The design study of Rai and Madavan[10] was repeated with the methodology described in this study. The initial design point was chosen to be the same.[10] The same three geometric parameters, i.e., the thickness values at the three control points, were used to generate

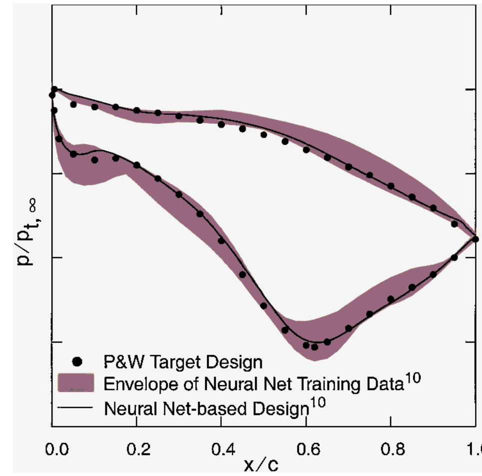


Fig. 6 Comparison of the target pressure distribution with that obtained from a CFD simulation for the neural net-based design of Rai and Madavan.[10] The shaded region represents the envelope of pressure distribution data obtained from CFD simulations that were used to train the neural net.

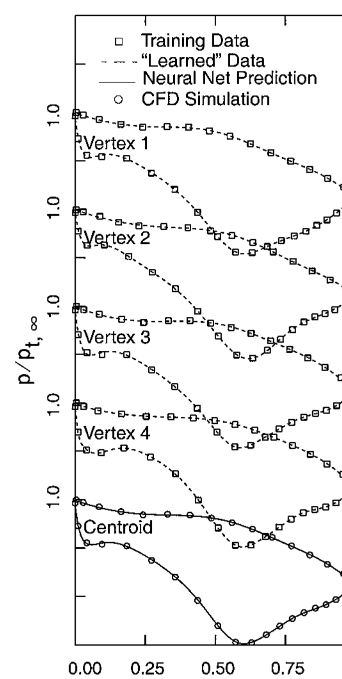


Fig. 7 Training the neural net. Comparison between training pressure data and net "learned" airfoil pressure distributions at the four vertices of the tetrahedron for the three-parameter design study. Airfoil surface pressure distributions predicted by the response surface are compared to CFD simulations at the centroid of the tetrahedron.

airfoil geometries. Parameter-based partitioning was accomplished in the following manner. Polynomial approximations were used to represent the variation of the airfoil surface pressure with the three geometric parameters. The neural net was used only to represent the variation of the pressure along the airfoil surface. For this kind of partitioning of the parameters, a linear variation results in a simple tetrahedron (four vertices) and a quadratic variation results in a tetrahedron with midside nodes (for a combined total of 10 vertices and nodes as in Fig. 3). A linear approximation was used here.

Each of the four three-layer nets (representing the four vertices of the tetrahedron) had two input nodes, one for the bias and one for the axial location, and one output neuron. The first and second hidden layers had 7 and 3 neurons, respectively, for a total of 38 connection weights. Thus, the total number of connection weights for all four nets was 152. During the training process the training error was reduced by about four orders of magnitude from the initial value.

Figure 7 shows the computed surface pressures (CFD) and those obtained from the neural net (learned pressures) at the four vertices of the simplex centred at the initial design point. The two sets of pressures are in good agreement. Figure 7 also shows the computed surface pressure distribution (CFD simulation) at the centroid of the simplex and that obtained from the response surface. The two sets of data are in good agreement and demonstrate the generalization capability of the composite response surface.

In Ref. 10, the optimal design lay within a cube of 27 points (CFD solutions) in design space. This was found not to be the case in the current approach. The volume of the tetrahedron, with its centroid positioned at the same initial design point,[10] is smaller than the volume of the cube and does not contain the optimum design point. Figure 8 shows the envelope of airfoil pressure distributions corresponding to the four vertices of the simplex (initial design point) and the target pressure. The target pressure distribution is not contained within this envelope. The optimization procedure together with the constraint that the search be limited to the volume of the tetrahedron converges to a point on one of the bounding surfaces of the simplex. Figure 8 shows that the pressure distribution at this point on the boundary of the simplex is in better agreement with the target distribution.

The process of constructing new simplexes and searching for the local optimum was repeated five times. Figure 9 shows the envelope of pressure distributions for the fifth (final) simplex that was constructed. In comparison with Fig. 8, the target pressure distribution is almost fully contained within the envelope in Fig. 9. The pressure distribution corresponding to the optimum design point is also shown in Fig. 9. A comparison of the optimal pressure distribution obtained in this three-parameter design study with that obtained in Ref. 10 shows that the two are nearly identical, with the present results showing some slight improvement. The improved results are probably because the volume of the tetrahedron bounding the optimal design point is smaller than the volume of the cube bounding this point in the earlier method,[10] thus resulting in a more accurate response surface.

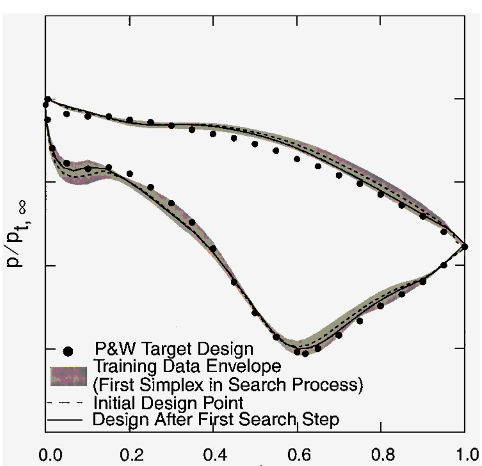


Fig. 8 Comparison of the initial airfoil pressure distribution with that obtained after the first search step for the three-parameter design study. The shaded region represents the envelope of pressure distribution data obtained from CFD simulations at the vertices of the first simplex and used to train the neural net.

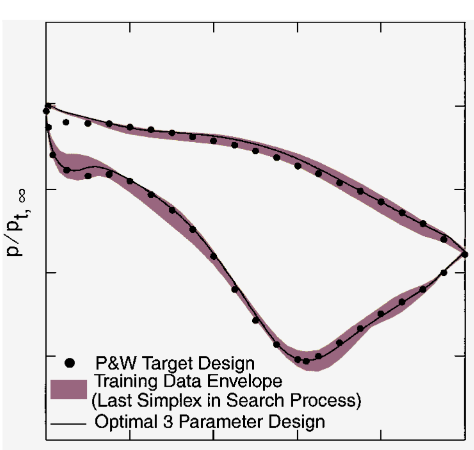


Fig. 9 Comparison of airfoil pressure distribution for the optimal three-parameter design with the target pressure distribution. The shaded region represents the envelope of pressure distribution data obtained from CFD simulations at the vertices of the last simplex and used to train the neural net.

Application to a Turbine Airfoil Design Involving Many Parameters

In the previous design studies, the number of geometric parameters was limited to three. This results in some differences between the optimal solution and the target. Many of the geometric parameters that were held constant in these initial designs must be allowed to vary to obtain a solution that is closer to the target. The current design procedure permits the use of a larger number of parameters with a modest increase in simulation time. To demonstrate the strength of this procedure the previous three-parameter design study was repeated with the following 15 geometric parameters: leading edge and trailing edge airfoil metal angles (two parameters), eccentricity of upper leading edge ellipse (one parameter), stagger angle (one parameter), axial location of maximum thickness value on the upper and lower surfaces (two parameters), airfoil thickness values at the leading edge (two parameters), and airfoil thickness values at intermediate points on the upper and lower surfaces (seven parameters).

Allowing these 15 geometric parameters to vary during the design process provides considerable flexibility in defining the airfoil. Note that some modifications were also made to the airfoil generation method (scheme A), e.g., defining the leading edge with two different ellipses. These modifications were made to provide more flexibility in defining the airfoil shape but are not discussed here in the interest of brevity.

The design study was repeated with these 15 parameters using a sequence of 16-vertex simplexes (linear variation within the simplex) in the search process. The optimal design obtained in the three-parameter study was used as the initial design point. Figure 10 compares the optimal design with that obtained earlier. Although both optimal designs are close to the target, the 15-parameter design is in better agreement along the pressure side and near the leading edge of the airfoil. The exit gas angle for the 15-parameter and 3-parameter designs was 21.4 and 21.8 deg, respectively, and agrees well with the target value of 21.5 deg. The two airfoil geometries corresponding to the current optimal design and the earlier design are compared in Fig. 11. As the corresponding pressure distributions shown in Fig. 10 indicate, the main differences between the two airfoils are on the pressure side of the airfoil from the leading edge to about 75% chord.

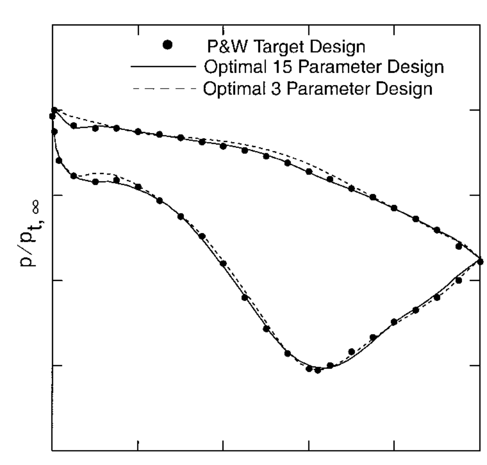


Fig. 10 Comparison of the airfoil pressure distributions obtained for the optimal 3- and 15-parameter designs.

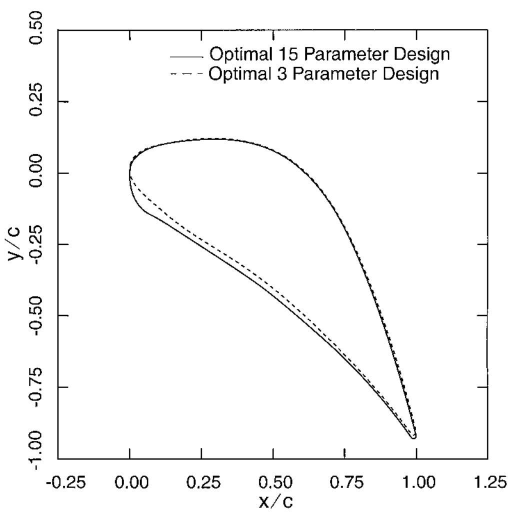
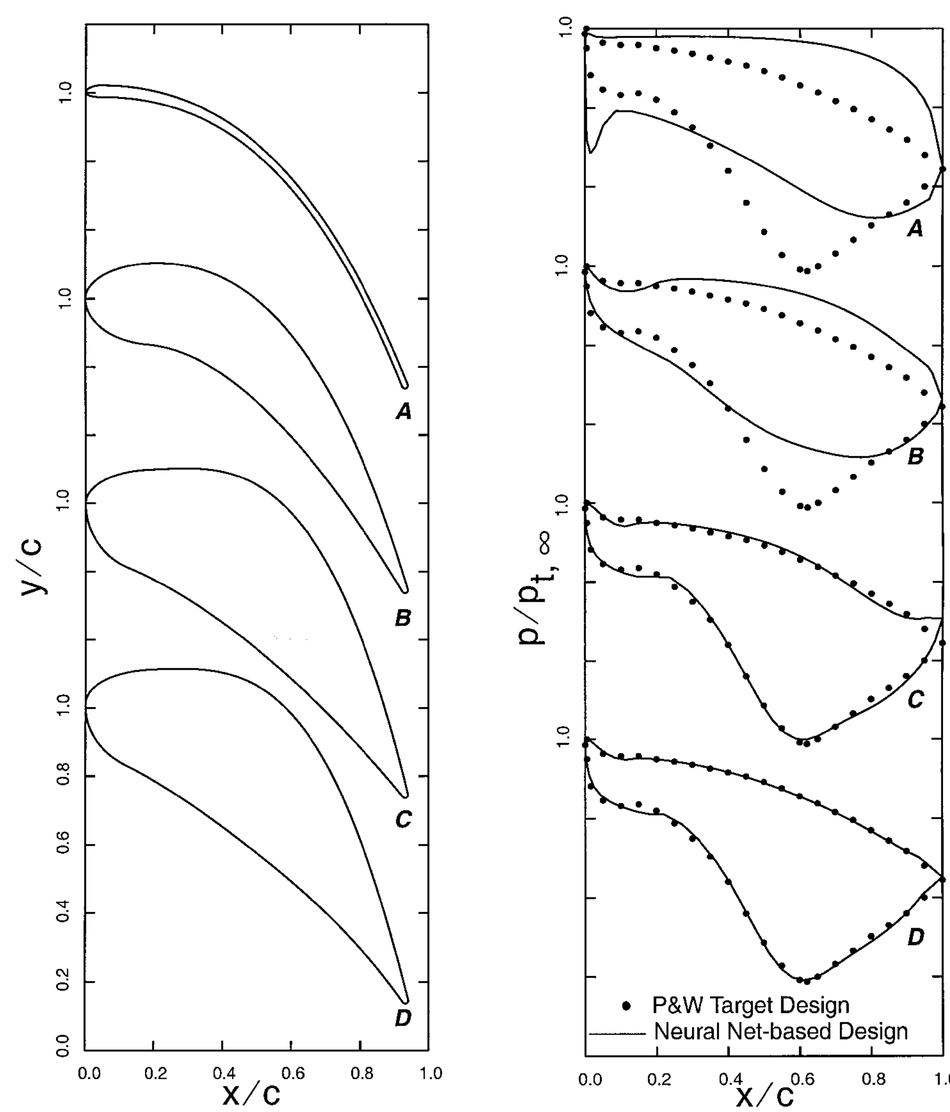


Fig. 11 Comparison of the airfoil geometries for the optimal 3- and 15-parameter designs.

Metamorphosis of a Generic Shape into an Optimal Airfoil

The initial design that was used in the three-parameter optimization study of Rai and Madavan[10] was obtained from a trial-and-error process and was fairly close to the final design. This was done so that the optimal design could be arrived at using a single response surface. The current methodology uses a sequence of response surfaces to enable a search of the design space thus permitting the use of initial designs that are far from the optimal design. To illustrate this capability, the initial geometry was chosen to be a generic curved plate. The inlet and exit metal angles for this curved plate were initially set equal to the corresponding gas angles. This initial geometry is shown in Fig. 12 by the airfoil marked A.

The current design methodology was then applied to obtain the optimal geometry. Figure 12a shows the progression of the airfoil geometry as the optimal design is approached. The corresponding pressure distributions are compared with the target pressure distribution in Fig. 12b. An additional feature that was incorporated into the design process was the use of CFD solutions of different fidelities. The design process was carried out using solutions to the Euler equations until the geometry denoted as C in Fig. 12a was obtained.



a) Airfoil geometry

b) Airfoil surface pressure distribution

Fig. 12 Transformation of the airfoil geometry from a simple curved plate to the optimal shape: A, initial design; B, midway through Euler optimization; C, start of Navier–Stokes optimization; and D, final design.

Subsequently, solutions to the Reynolds-averaged Navier–Stokes equations were used to achieve the final design shown as airfoil D in Fig. 12a.

Using the mean camber line and thickness distribution to obtain the airfoil geometry (scheme A) was found to be cumbersome and prone to producing inadmissible airfoil geometries during the optimization process. The new approach (scheme B) was used instead and found to be robust even though the initial design was far from optimal. In the initial stages of design only two parameters were varied at each optimization step to obtain the response surface. Three sets of such two-parameter combinations were used in sequence for the initial development of the airfoil. These sets of two-parameter combinations were obtained after some experimentation and should be generally applicable to a wide range of airfoil shapes. As a general rule, one should keep in mind that the incorporation of as much domain-specific information as possible into the search and optimization process will help to accelerate the evolution of the design. The three sets of two-parameter combinations used in this study are 1) the angles α_u and α_l at the leading edge, 2) the angles β_u and β_l at the trailing edge, and 3) the leading edge thickness values t_u and t_l.

After a reasonable airfoil shape was obtained, all six of the preceding parameters were used together to construct the response surface for a few more optimization steps. Additional parameters such as the leading and trailing edge metal angles, eccentricity of the upper leading edge ellipse, and other control points used to define the upper and lower tension splines were included in the final stages of design. The final design was obtained with a total of 13 geometric parameters and is shown as airfoil D in Fig. 12a. This design exercise demonstrates the ability of the method to transform a generic shape that is far from optimal into the final optimal design.

Figure 13 shows the variation of the mean square error (MSE) in surface pressure as a function of the cumulative computing time (on a single processor Cray C-90) used to generate the CFD data for the sequence of response surfaces used during the optimization process. This error is defined as

$$\mathrm{MSE} = \sum_{i=1}^{imax} \frac{(\tilde{p}_i - p_i)^2}{imax} \tag{7}$$

where $\tilde{p}_i$ is the target pressure, p_i is the pressure at the same axial location for any particular airfoil generated during the design process, and *imax* is the total number of airfoil surface points at which the target pressure is defined.

The time used to train the nets is not included in Fig. 13 because it is a small fraction of the total CPU time. Figure 13 also shows the time at which the switch from the Euler equations to the Navier–Stokes equations was made. Very little change is seen after approximately 40 CPU hours, indicating that for all practical purposes the design had converged. It should be noted that the

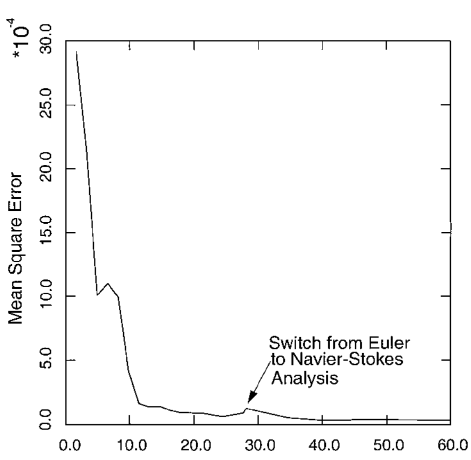


Fig. 13 Reduction of the mean square error in surface pressure with the cumulative computing time used in obtaining the CFD solutions. CPU time in single processor Cray C-90 hours.

preceding sequencing (three sets of two parameter airfoil families for the initial transformation of the curved plate) was obtained after some experimentation. This sequence results in a robust design process that can be used to generate a variety of airfoils.

Summary and Conclusions

A method for aerodynamic design that incorporates the advantages of both traditional RSM and neural networks has been developed. The procedure employs a strategy called parameter-based partitioning of the design space and uses a sequence of response surfaces based on both neural networks and polynomial fits to traverse the design space in search of the optimal solution. This approach results in response surfaces that have both the power of neural networks and the economy of low-order polynomials (in terms of number of simulations needed and network training requirements). Such an approach can handle design problems with many more parameters than would be possible using neural networks alone.

As in conventional RSM, the current design process permits the designer to perform a variety of tradeoff studies rapidly. Once the final response surface has been constructed using a certain envelope of pressure distributions, new design targets that lie within this envelope can be obtained very rapidly. This is not true of a brute force optimization technique because it would resort to function evaluations (CFD simulations) for every new design target. The ability to obtain new design targets rapidly is a major advantage because designers routinely perform a range of tradeoff studies before arriving at the final design.

Another advantage of the current approach is that it allows the use of low-fidelity simulations in the early stages of the design and a smooth transition to higher fidelity simulations as the search for the optimal design evolves. This can significantly reduce the computational costs incurred in simulation-based design.

The method developed here has been applied to the redesign of a turbine airfoil from a modern P&W jet engine. This design involved the use of 15 geometric parameters. The results obtained are closer to the target design than those obtained in an earlier study using only three design parameters. The capability of the method in transforming generic shapes, such as simple curved plates, into optimal airfoils has also been demonstrated.

The process developed here is extendable to three-dimensional geometries. In fact, although the use of 15 geometric parameters to define a two-dimensional turbine airfoil geometry may seem excessive, it does indicate that the current method has potential application to large-scale three-dimensional aerodynamic design.

Acknowledgment

A patent application that covers some of the original ideas in this paper has been filed by NASA.

References

[1]Fan, X., Herbert, T., and Haritonidis, J. H., “Transition Control with Neural Networks,” AIAA Paper 95-0674, Jan. 1995.

[2]Kawthar-Ali, M. H., and Acharya, M., “Artificial Neural Networks for Suppression of the Dynamic Stall Vortex over Pitching Airfoils,” AIAA Paper 96-0540, Jan. 1996.

[3]Faller, W. E., and Schreck, S. J., “Unsteady Fluid Mechanics Applications of Neural Networks,” AIAA Paper 95-0529, Jan. 1995.

[4]Norgaard, M., Jorgensen, C. C., and Ross, J. C., “Neural Network Prediction of New Aircraft Design Coefficients,” NASA TM 112197, May 1997.

[5]Meade, A. J., “An Application of Artificial Neural Networks to Experimental Data Approximation,” AIAA Paper 93-0408, Jan. 1993.

[6]Greenman, R., and Roth, K., “High-Lift Optimization Design Using Neural Networks on a Multi-Element Airfoil,” American Society of Mechanical Engineers, ASME Paper DETC98/CIE-6006, Atlanta, GA, Sept. 1998.

[7]Hajela, P., and Berke, L., “Neural Networks in Structural Analysis and Design: An Overview,” *Proceedings of the AIAA/USAF/NASA/OAI 4th Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Washington, DC, 1992, pp. 902–914.

[8]Huang, S. Y., Miller, L. S., and Steck, J. E., “An Exploratory Application of Neural Networks to Airfoil Design,” AIAA Paper 94-0501, Jan. 1994.

[9]Sanz, J. M., “Development of a Neural Network Design System for Advanced Turbo-Engines,” Fourth U.S. National Congress on Computational Mechanics, San Francisco, CA, Aug. 1997.

[10]Rai, M. M., and Madavan, N. K., “Application of Artificial Neural Networks to the Design of Turbomachinery Airfoils,” AIAA Paper 98-1003, Jan. 1998.

[11]Montgomery, D. C., *Design and Analysis of Experiments*, Wiley, New York, 1997, Chap. 14.

[12]Myers, R. H., and Montgomery, D. C., *Response Surface Methodology—Process and Product Optimization Using Designed Experiments*, Wiley, New York, 1995.

[13]Guinta, A. A., Golividov, O., Knill, D. L., Grossman, B., Mason, W. H., and Watson, L. T., "Multidisciplinary Design Optimization of Advanced Aircraft Configurations," *Fifteenth International Conference on Numerical Methods in Fluid Dynamics*, Vol. 490, Lecture Notes in Physics, edited by P. Kutler, J. Flores, and J. J. Chattot, Springer–Verlag, Heidelberg, Germany, 1997, pp. 14–34.

[14]Rai, M. M., and Madavan, N. K., "Multi-Airfoil Navier–Stokes Simulations of Turbine Rotor-Stator Interaction," *Journal of Turbomachinery*, Vol. 112, July 1990, pp. 167–190.

[15]Barron, A. R., "Approximation and Estimation Bounds for Artificial Neural Networks," *Machine Learning*, Vol. 14, Jan. 1994, pp. 115–133.

[16]Drela, M., "Elements of Airfoil Design Methodology," *Applied Computational Aerodynamics*, edited by P. A. Henne, Vol. 125, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1990, pp. 167–189.

[17]Volpe, G., "Inverse Airfoil Design: A Classical Approach Updated for Transonic Applications," *Applied Computational Aerodynamics*, edited by P. A. Henne, Vol. 125, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1990, pp. 191–220.

[18]Dring, R. P., and Heiser, W. H., "Turbine Aerodynamics," *Aerothermodynamics of Aircraft Engine Components*, edited by G. C. Oates, AIAA, New York, 1985, pp. 219–271.

A. D. Belegundu
Associate Editor