JOURNAL OF PROPULSION AND POWER
Vol. 17, No. 1, January–February 2001

Application of Artificial Neural Networks to the Design of Turbomachinery Airfoils

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

The feasibility of applying artificial neural networks to the aerodynamic design of turbomachinery airfoils is investigated. The design process involves defining a target pressure distribution, computing several flows to adequately populate the design space in the vicinity of the target, training the neural network with this data, and, finding a design that has a pressure distribution that is closest to the target. The last step is carried out using the network as a function evaluator. This design process is tested using an established flow simulation procedure, a simple two-layer feedforward network and a conjugate gradient optimization technique. Results are presented for some validation tests as well as a complete design effort where the pressure distribution from a modern Pratt and Whitney turbine was used as a target. These results are very encouraging and clearly warrant further development of the process for full three-dimensional design.

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 interpolation purposes, and grid generation. Fan et al.[1] studied the possibility of achieving active laminar flow control using "smart walls," a combination of neural networks and microelectromechanical devices. Kawther-Ali and Acharya[2] demonstrated the feasibility of using neural networks in the control of dynamic stall. Neural networks have also been used to both model unsteady flows and to optimize aerodynamic performance parameters.[3] The feasibility of reducing wind tunnel test times for a given configuration by using neural networks 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 networks has been presented by Meade.[5] Rodi[6] used neural networks designed to prevent grid folding to enhance grids obtained with a hyperbolic grid generation scheme for surfaces with concave curvature. Neural network applications in aeronautics are not limited to aerodynamics. An excellent review of the applications of a variety of neural networks in structural analysis and design is provided by Hajela and Berke.[7]

In this paper, we explore the feasibility of applying neural networks to aerodynamic design, in particular, the design of turbomachinery airfoils. Prior work in this general area includes a preliminary effort at airfoil design using neural networks[8] where lift, drag, and pitching moment coefficients obtained from an inverse design code are used to train a feedforward neural network. The network is then used to predict the variation of these 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 another work,[9] a neural network is trained to extract the required input pressure distribution from a database of pressure distributions and an inverse-design method is then used to compute the airfoil shape that corresponds to this "network" input pressure distribution. The neural network is therefore used to determine the pressure distribution that would produce the required flow conditions.

The focus of the current effort is on developing an intelligent design process that uses existing data (or perhaps data that will be generated as required). The attempt is to develop an aide to the designer, one that fully utilizes the ability of the computer to interrogate or interpolate unstructured data in multiple dimensions—a task that is difficult for humans to perform. In addition, such an aide should help the designer to rapidly perform a variety of trade-off studies.

The principal idea behind the current effort is to represent the objective function in 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 exhibits optimal performance characteristics. The basic design process is conceptually similar to the ones that are used in structural design.[7]

While other interpolation methods can also be used to represent the design data, neural networks are particularly suitable for multidimensional interpolation where the data are not structured (for example, on arbitrary grids). Since most design problems in aerodynamics involve a multitude of parameters and, additionally, unstructured datasets, neural networks may 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 typically 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 of neural networks indicates that they provide a natural framework within which a succession of solutions of increasing fidelity can be represented

Presented as Paper 98-1003 at the 36th Aerospace Sciences Meeting and Exhibit, Reno, NV, 12–15 January 1998; received 15 August 1999; revision received 13 March 2000; accepted for publication 22 March 2000.

*Senior Scientist, M/S 269-1, Information Sciences and Technology Directorate; mrai@mail.arc.nasa.gov.

†Research Scientist, M/S T27A-1, Numerical Aerospace Simulation Systems Division; madavan@nas.nasa.gov.

and subsequently utilized for optimization and design. Of particular importance is the fact that the design data can come from a variety of sources, including experiments and computations. Rules-of-thumb that expert 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 engine industry, which has accumulated enormous amounts of experimental data and numerous design rules over a number of decades.

In this paper we explore the possibility of designing turbine airfoils using CFD data, a simple two-layer feedforward neural network, and a conjugate-gradient-based optimization routine. Two-dimensional rotor and stator airfoil cross sections are used to develop the concept. The design process developed here is then validated and tested by applying it to the design of the midspan geometry of a turbine airfoil. The pressure distribution from a modern Pratt and Whitney (P&W) turbine is used as the desired target. The computed surface pressure distribution for the resulting airfoil geometry is found to be in excellent agreement with the target distribution. Details of the process and the results from validation tests, as well as the P&W design study, are presented in this paper.

CFD Methodology

The ROTOR-2 computer program[10] 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 1) the pressure ratio across the turbine (ratio of exit static pressure to inlet total pressure), 2) the inlet temperature and flow angle, and 3) the unit Reynolds number based on inlet conditions. Details regarding the solution methodology can be found in Ref. 10.

Neural Network

The simple two-layer feedforward neural network shown in Fig. 1 is trained using the airfoil data (surface pressures nondimensionalized by the inlet total pressure). The pressure values are obtained from CFD simulations as a function of the distance along the airfoil surface from the leading edge. The first node in the input layer is a bias node (input of 1.0). The second input node accepts the normalized arc-length values for the various points on the airfoil surface. The remaining nodes accept the geometry parameters for that particular airfoil simulation. Pressure values corresponding to the arc-length inputs to the second node in the input layer are obtained at the output node.

The network is trained using the computed pressures, surface grid point distributions, and the geometry parameter. The trained network (defined by the network structure and weights) represents the variation of pressure as a function of the geometry parameters and the location on the airfoil surface. The accuracy of this representation is determined by the accuracy of 1) the CFD simulations, 2) the population density of the simulations used to populate the design space, and 3) the network parameters such as the number of neurons in the hidden layer (Fig. 1). The accuracy of the CFD simulations will not be discussed here. The accuracy with which the net represents the training data is given by the objective function that is minimized to obtain the weights

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

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 grid points on the surface of the airfoil, and $nmax$ is the number of CFD solutions used to train the network. The number of neurons is increased successively until the objective function is a sufficiently small number. In this study, the objective function was reduced by about four orders of magnitude for each of the validation cases. This reduction in the objective function was deemed adequate (results are presented in the next section).

The accuracy with which the network represents the objective function can be estimated as follows:

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

2) Evaluate the pressure distributions for these same cases using the neural network.

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

Such a verification process is essential in order to establish the adequacy of the generalization capabilities of the network for the given network architecture and pressure data.

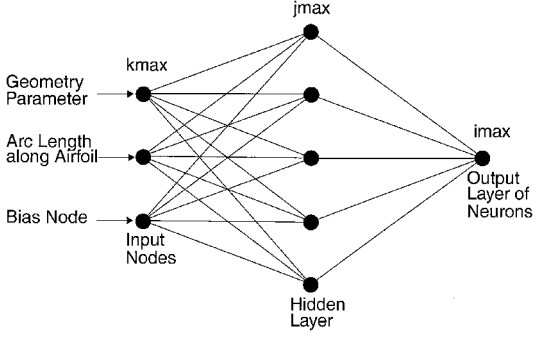

Fig. 1 Schematic of a two-layer feedforward neural network.

Design Methodology

The overall strategy of using artificial neural networks to design an airfoil with a prescribed pressure distribution can be summarized as follows:

1) Define the target pressure distribution.

2) Compute several CFD solutions such that they adequately populate the design space in the vicinity of the design point (a rough guess).

3) Train the network using these datasets.

4) Using the network as a function evaluator, find a design that has a pressure distribution that is closest to the target pressure distribution.

In the second step of the design process outlined above, CFD solutions are computed for a range of input geometric parameters such that the corresponding pressure distributions encompass the target. In other words, the target pressure distribution must be contained within the pressure distributions that constitute the training set—a necessary, but not sufficient, condition for the current design process to work. An initial geometry with a corresponding pressure distribution that is close to the target first determined. The geometric parameters for this geometry are then perturbed by equal positive and negative amounts to obtain the training set. The further away the initial geometry is from the target, the larger the perturbations that are required to ensure that the training set encompasses the target. In the design effort reported here, the initial point was obtained by a trial-and-error process using a Navier–Stokes code. Note that there are alternative approaches, such as inverse design procedures[11,12] and meanline analyses and streamline curvature-based methods[13] that can also be used to obtain the initial design point.

The design process described above is clearly an iterative one. In the first iteration one would like to make the region of design space that is represented by the network as large as possible so that the initial design point does not have to be very close to the target. The design space is enlarged by allowing wide variations in the airfoil geometry and, in turn, requires a large number of CFD solutions to adequately populate the design space. The need to minimize overall design costs dictates that these solutions be obtained at relatively low cost (e.g., potential flow solutions). Once an approximate design point is obtained, the region around this point can be populated with more refined computations (e.g., Euler

or Navier–Stokes solutions), and the network retrained for a second iteration. The retraining should not be prohibitively expensive since the design space can now be much smaller in extent than before, and hence require fewer solutions to adequately populate. Additional refinement of the design point can be obtained by another iteration where solutions of the same fidelity (e.g., Navier–Stokes solutions) are computed very close to the predicted design point.

The final step in the process is a validation step where the geometry corresponding to the design point (obtained in step 4) 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.

This design process consists of two important steps, namely, training the network, and then searching the design space that is now represented by the network to find the optimal design. Our experience with aerodynamic design has been that, of these two steps, the former is computationally more intensive by two or three orders of magnitude. One important aspect of the current methodology is that design constraints, rules-of-thumb, and other requirements are specified in the second, much less expensive, step. Specifying the requirements in the second step allows the designer to perform rapidly a variety of trade-off studies that would naturally involve changing the constraints to resolve design conflicts.

Generation and Interpolation of Turbine Airfoil Geometries

The ability to represent several airfoil geometries with a common set of geometrical parameters is essential to this design process. The parameters must be chosen such that variations or combinations of the set of airfoil geometries that are used to train the network can be obtained by smoothly varying these parameters. Geometrical constraints imposed for structural reasons, aerodynamic reasons (e.g., to eliminate flow separation), etc., 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.

The midspan stator and rotor geometries reported in the experimental study of Dring et al.[14] were used for the purpose of validating the design process. Each airfoil was first parameterized using its mean camber line and thickness distribution. The thickness of the airfoil and the coordinates of the mean camber line were obtained as functions of the normalized arc-length along the camber line, ξ. The arc-length was normalized by the total arc-length of the camber line.

In addition to the rotor and stator airfoil geometries, a third geometrical element—a horizontal straight line—was used. This element was included to provide more control over the inlet and exit angles of the interpolated airfoils by reducing the camber of the airfoil. The geometric parameters, R, S, and L, were used to represent the rotor, stator, and horizontal line, respectively. Each triad of R, S, and L values represents an airfoil geometry. These parameters satisfy the following constraints:

$$R + S + L = 1 \tag{2}$$

$$0 \leq R, S, L \leq 1 \tag{3}$$

Figure 2 is a geometrical representation of the interpolation process used to obtain variants of the primary geometric elements. Each vertex represents a particular geometrical element, and all other points on or within the triangle represent interpolated geometries. The geometry parameters R, S, L for the interpolated point represented by the solid symbol on Fig. 2 are given by

$$R = A_1/A, \qquad S = A_2/A, \qquad L = A_3/A \tag{4}$$

where A_1, A_2, and A_3 are the areas of the triangles shown in Fig. 2, and A is the area of the entire triangle ($A = A_1 + A_2 + A_3$).

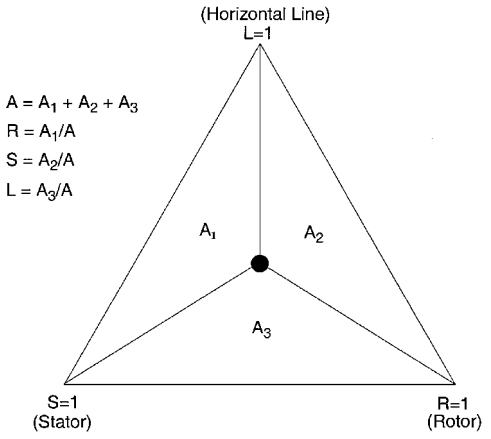

Fig. 2 Geometric representation of the interpolation process used to obtain the different airfoil cross sections used in this study. The solid symbol denotes the interpolated geometry that is being generated.

Variations of the elemental geometries were obtained as follows:

$$X = SX_{\text{stator}} + RX_{\text{rotor}} + LX_{\text{line}} \tag{5}$$

$$Y = SY_{\text{stator}} + RY_{\text{rotor}} + LY_{\text{line}} \tag{6}$$

$$T = \frac{ST_{\text{stator}} + RT_{\text{rotor}}}{S + R} \qquad \text{for} \qquad S + R > 0 \tag{7}$$

where X, Y represent the coordinates of the mean camber line of the airfoil, T is the airfoil thickness function, and X, Y, and T are all functions of ξ.

Validation of Design Methodology

Validation tests of the design methodology described above were performed to obtain a better understanding of the ability of the neural network to represent complex aerodynamic data of the type normally encountered in turbomachinery design. These validation tests demonstrate the ability of the design process to determine an interpolated variant of a given set of airfoils that yields a pressure distribution that is closest (in a root mean square sense) to a target pressure distribution subject to certain constraints. The basic turbine airfoil geometries were obtained from an earlier experimental investigation of rotor-stator interaction in a low-speed axial turbine.[14] The airfoil generation and interpolation procedure was used to obtain airfoil sections for training the network. Two sets of tests were conducted: Case A, where only one geometric parameter was needed to specify the airfoils, and Case B, where two geometric parameters were needed to specify the airfoils.

Case A: Single-Parameter Optimization

In this case the airfoil sections used to train the network were obtained by interpolating between the stator and rotor airfoils ($L = 0$). The training datasets and the target are graphically represented in Fig. 3. Note that interpolation and optimization for this case is along the lower side of the triangle because the horizontal line is not used ($L = 0$). Thus, only a single geometric parameter is needed to prescribe an airfoil geometry. Pressure distributions were obtained for the cases $S = 0.0$, 0.25, 0.5, 0.75, and 1.0 with the ROTOR-2 computer program.[10] The network was trained with 3 input nodes and 1 output neuron. During the training process, the number of neurons in the hidden layer was gradually increased until the objective function was reduced by about four orders of magnitude from the initial value. For this particular case, 31 neurons were required in the hidden layer to achieve this level of reduction in the objective function. Figure 4 shows the airfoil geometries corresponding to the $S = 0.0$, 0.5, and 1.0 cases. Figure 5 compares the training pressure data and the data obtained using the neural network (learned airfoil surface pressures) for the cases $S = 0.0$, 0.5, and 1.0. Although not shown, similar comparisons between the training pressure data and

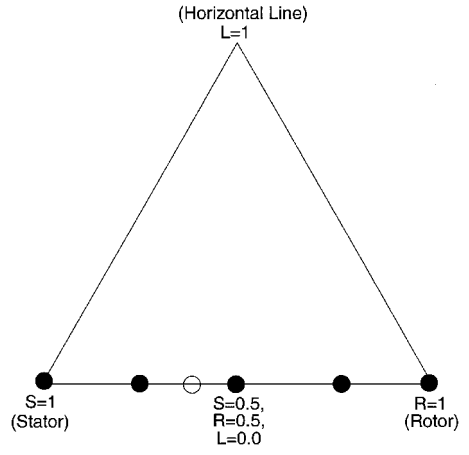

Fig. 3 Geometric representation of the interpolation process used to obtain the different airfoil cross sections used in the Case A optimization study. Solid symbols represent datasets used to train the network. The open symbol represents the test case used to validate the network.

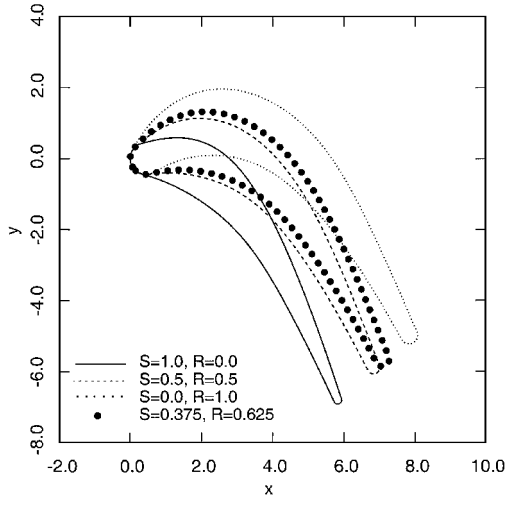

Fig. 4 Airfoil geometry variation obtained by interpolating between the $S = 1.0, R = 0.0$ and $S = 0.0, R = 1.0$ airfoil cases.

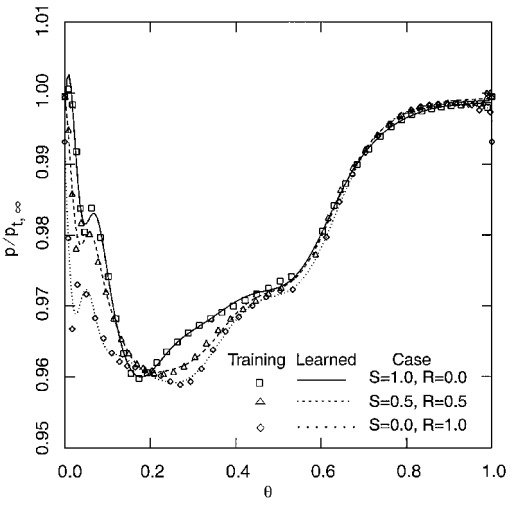

Fig. 5 Training the neural network. Comparison between training pressure data and network "learned" airfoil surface pressure distributions.

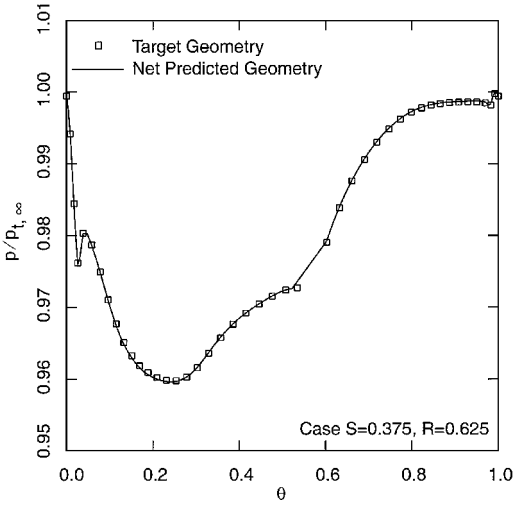

Fig. 6 Comparison of the airfoil surface pressure distributions obtained by CFD simulations for the target airfoil geometry and the geometry predicted by the neural network.

those from the neural network were obtained for the cases $S = 0.25$ and 0.75. The two sets of pressures are in good agreement, although minor differences are noted in the suction-side overspeed region. These differences can be eliminated by continuing the training process with an adequate number of neurons in the hidden layer. In this validation study, it was found that resolving these smaller features very accurately was not necessary to obtain a good design. However, this may not be true in all cases, and some designs may require a high level of accuracy in the representation of the training data. Note that in Fig. 5 (and in subsequent figures showing surface pressure distributions), the surface pressures are normalized by the average inlet total pressure $p_{t,\infty}$ and are plotted as a function of the normalized arc-length along the airfoil surface, θ.

The neural network was then used in a design mode. A CFD simulation was conducted for the case $S = 0.375$ and used as the target in the design process. The airfoil cross section corresponding to this is also shown in Fig. 4. The design value of S obtained using the neural network and optimizer was 0.39. Figure 6 compares the target distribution and the pressure distribution obtained from a CFD simulation with the geometry parameter $S = 0.39$. The two pressure distributions are in good agreement, thus validating the optimal design point predicted by the neural network.

Case B: Multiparameter Optimization

Case A dealt with a situation where a single geometric factor was sufficient to completely specify the geometry. Interpolation and optimization were along the lower side of the triangle shown in Fig. 3. In Case B, two geometric parameters (S and R) are required to determine the airfoil geometry. The design space now includes the interior of the triangle in Fig. 3 and is not limited to a single side as in Case A. Case B is thus more demanding of the design process than Case A.

Figure 7 shows the matrix of airfoil geometries used to train the network. The neural network was trained with the data from 15 airfoil geometries (1324 data points) using four input nodes, one output neuron, and 91 neurons in the hidden layer. The target pressure distribution was obtained with the parameters $S = 0.350$ and $R = 0.600$. The network predicted values for S and R were 0.277 and 0.675, respectively. These values of S and R are significantly different from those used to generate the target pressure distribution. The computed pressure values for the airfoil geometry corresponding to these geometric parameter values are compared with the target pressure distribution in Fig. 8. Surprisingly, there is good agreement between the two pressure distributions.

This unexpected behavior can be better understood by studying the objective function. Figure 9a shows contours of the objective function that is minimized [see Eq. (1)] within the design space (lower portion of the triangle represented as ABCD in Fig. 7) to obtain the optimal values for S and R. The contours show a large low gradient region surrounding the minimum point. This results in a situation where relatively large variations in the design parameters

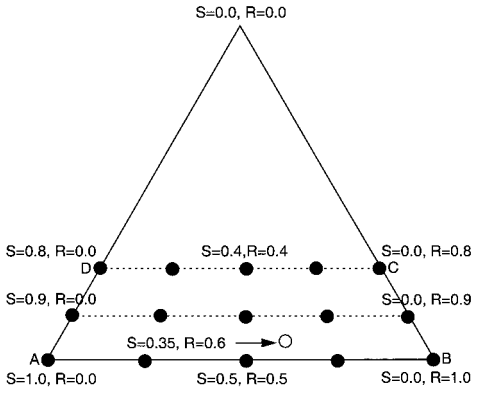

Fig. 7 Geometric representation of the airfoil cross-sections used in the Case B optimization study. Solid symbols represent datasets used to train the network. The open symbol represents the test case used to validate the network.

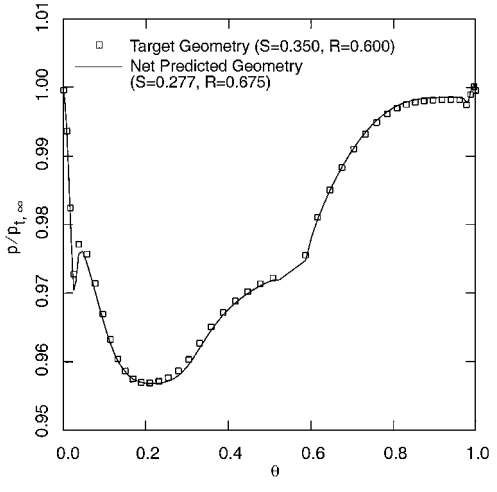

Fig. 8 Comparison of the airfoil surface pressure distributions obtained by CFD simulations for the target airfoil geometry and the geometry predicted by the neural network.

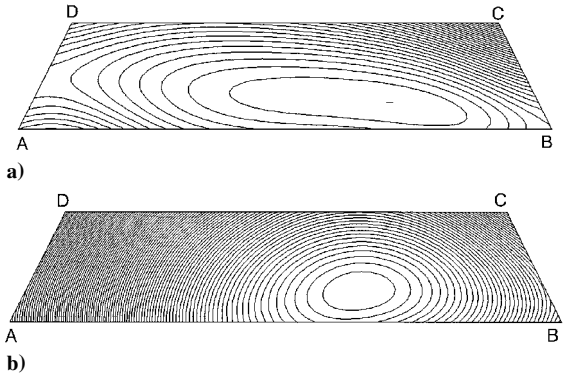

Fig. 9 Contours of the objective function in the allowed design space. a) Original objective function; b) modified objective function (modified to include inlet angle constraint). Contours in the range 0.620E-04 to 0.217E-01 in steps of 0.216E-03.

yield essentially the same pressure distribution. The ultimate choice of the optimal design point depends on the accuracy with which the network represents the design space. As mentioned earlier, this, in turn, depends on the population density of the simulations used to populate the design space and the network parameters such as the number of neurons in the hidden layer.

Although the computed pressures for the airfoil corresponding to the design point agree well with the target pressure distribution as

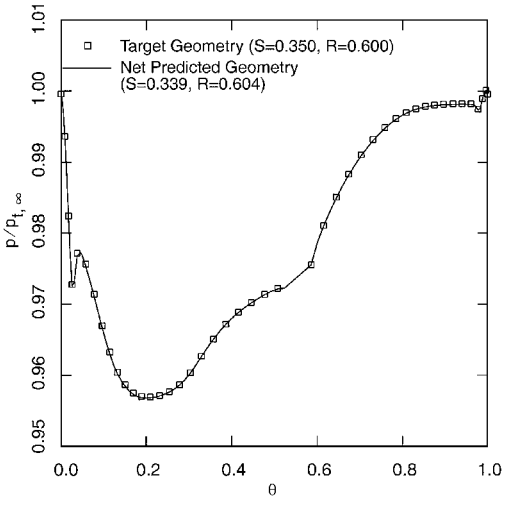

Fig. 10 Comparison of the airfoil surface pressure distributions obtained by CFD simulations for the target airfoil geometry and the geometry predicted by the neural network with the inlet angle constraint.

seen in Fig. 8, the angle made by the camber line at the inlet with the y-axis (α^{inlet}) was found to be different than that for the target airfoil (by about 3.5 deg). To obtain the correct inlet angle a penalty function approach was used. The modified objective function is given as

$$f = \lambda\left(\alpha^{\text{inlet}}_{\text{target}} - \alpha^{\text{inlet}}_{\text{design}}\right)^2 + \sum_{n=1}^{nmax}\sum_{i=1}^{imax}\left(\tilde{p}_i^n - p_i^n\right)^2 \qquad (8)$$

where λ is a constant. Minimization of the modified objective function yielded $S = 0.339$ and $R = 0.604$, which are much closer to the target values of 0.350 and 0.600, respectively. Contours of the modified objective function are shown in Fig. 9b. The addition of the constraint changes the contours significantly and results in a sharply defined optimal point. Figure 10 shows the computed pressures for the new optimal design. A slight improvement is noticed over the previous design both on the pressure and suction sides. Additionally, the leading edge angular difference is reduced to less than 0.2 deg. The difference in trailing edge angles was less than 0.35 deg in both cases.

This exercise of introducing a constraint was performed primarily to illustrate the use of penalty functions in incorporating constraints in the neural network-based design methodology. It was also performed to demonstrate how the computed optimal solution could be brought closer to the real optimum. Such an analysis would not be required in a practical situation. For example, in the context of the current problem, the geometry obtained in the unconstrained optimization ($S = 0.277$, $R = 0.675$) is "optimal" because it produces a pressure distribution that is sufficiently close to the target pressure distribution.

Turbine Airfoil Design Application

Cases A and B represent validation tests that demonstrate the ability of the design process developed here to determine an interpolated variant of a given set of airfoils that yields a pressure distribution that is closest to a target pressure distribution (in a root mean square sense), subject to certain constraints. To test the efficacy of this design process further, a real turbine airfoil design was attempted.

The target pressure distribution was supplied by Frank Huber and Richard Rowey of Pratt and Whitney (P&W). This pressure distribution was obtained at the midspan of a turbine vane from a modern P&W jet engine. The following flow and geometry parameters were also supplied 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 number

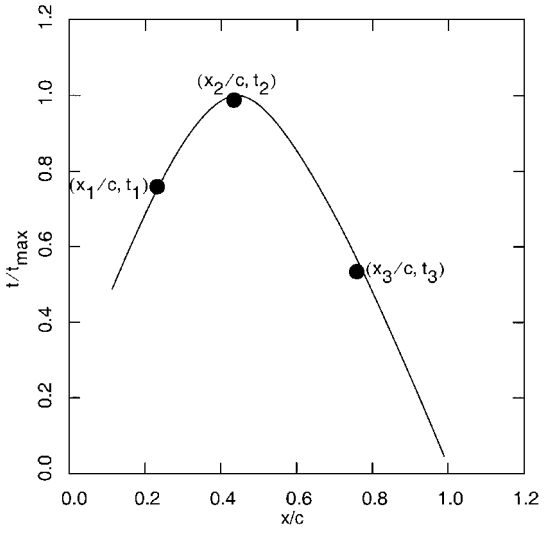

Fig. 11 Turbine airfoil thickness distribution as a function of the axial distance along the airfoil.

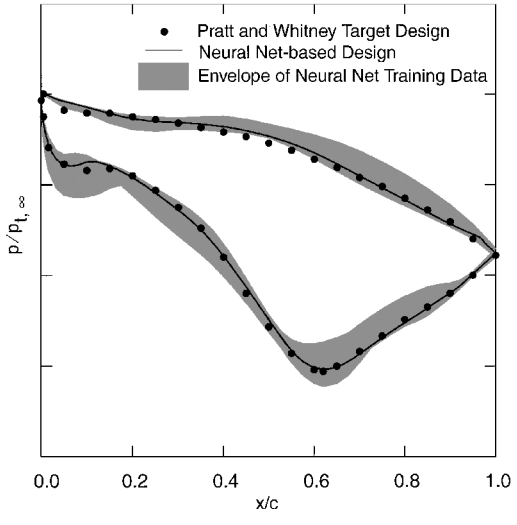

Fig. 12 Comparison of the airfoil surface pressure distributions obtained by CFD simulations for the neural network-based design and the target Pratt-and-Whitney design. The shaded region represents the envelope of pressure distribution data obtained from CFD simulations that were used to train the neural network.

of vanes in the row. The actual airfoil geometry was not provided because this design effort was intended to be a blind test.

Initiating the design process requires the generation of several airfoils whose pressure distributions form 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. The thickness distribution for this airfoil is shown in Fig. 11. This distribution was represented by a tension spline that requires 1) the location of the maximum thickness (normalized by the axial chord c) and its value (x_2/c, t_2), 2) the thickness value midway between the point of maximum thickness and the start of the camber line (t_1), and 3) the thickness value midway between the point of maximum thickness and the end of the camber line (t_3). The thickness values at the ends of the camber line were specified a priori (as provided by Frank Huber and Richard Rowey). The parameters t_1, t_2, and t_3 were varied to obtain the training set (three values of each). A total of 27 airfoil geometries and corresponding pressure distributions were computed. Figure 12 shows the envelope of airfoil pressure distributions obtained by CFD simulations for this set of airfoils.

The neural network was trained using these 27 pressure distributions, five input nodes (one bias node, one node to represent the position on the airfoil and one node each for the three geometric

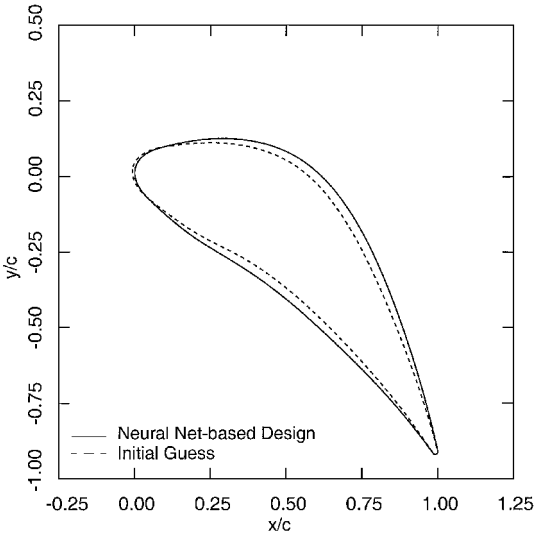

Fig. 13 Comparison of initial guess and final neural network-based airfoil geometry.

parameters t_1, t_2, and t_3), 91 neurons in the hidden layer, and one output node. Figure 12 also shows the target pressure distribution and the pressure distribution obtained by the CFD simulation for the optimal design predicted using the neural network. The two are in good agreement. Figure 13 is a comparison of the final design airfoil geometry with the initial guess of the geometry. This initial guess represents the initial design point that was perturbed to obtain the neural network training datasets. The exit gas angle and Mach number for the neural network-based design compared well with that for the target design. The exit gas angle was 21.5 deg and 21.7 deg for the target and neural network-based design, respectively, whereas the exit Mach number was 0.571 and 0.565 for the target and neural network-based design, respectively.

Convergence Studies for the CFD Simulations

The CFD simulations for the neural network-based design of the P&W airfoil were performed on composite grids with about 11,500 grid points (151×41 points in the inner grid, and 121×41 points in the outer grid). All the simulations were carried out to the point where the maximum residual in the density was about 10^{-5}. To check the accuracy of the CFD simulations, grid and residual convergence studies were conducted for the final neural network-based design. To demonstrate residual convergence, the CFD simulation was further converged to near machine accuracy with the maximum residual in the density reduced to 10^{-12}. To demonstrate grid convergence, the CFD simulation was repeated on a grid with almost twice the number of grid points in each direction (301×71 points in the inner grid, 221×71 points in the outer grid, for a total of 37,000). The pressure distributions obtained from these convergence studies are shown in Fig. 14. The pressure distributions compare well with that for the neural network-based design simulation, and demonstrate that the residual convergence criteria and grid densities used to obtain the results in the paper are adequate to resolve the essential features of the flowfield.

Accelerating the Neural Network Training Process

The P&W design case described here required about 27 hours of central processing unit (CPU) time to generate the CFD solutions and about 2 hours to train the network on a single processor CRAY-C90. It is estimated that the CFD code could be made more efficient using simpler flux evaluations and improved linearization techniques. This could reduce the CPU time requirements by a factor of two or three. In addition, the use of lower-fidelity CFD solutions in the early design phase would greatly reduce the amount of CPU time required for simulations (for example, potential flow-based CFD solutions for the same P&W design case could have been accomplished in a few minutes). Once this happens, the time required

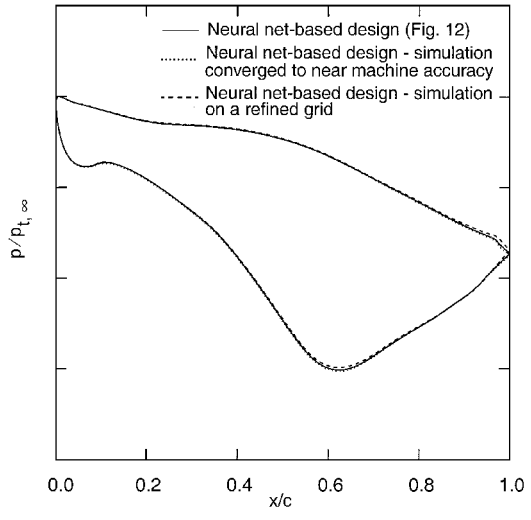

Fig. 14 Comparison of the airfoil surface pressure distributions obtained from grid and residual convergence studies of the CFD simulations for the neural network-based design.

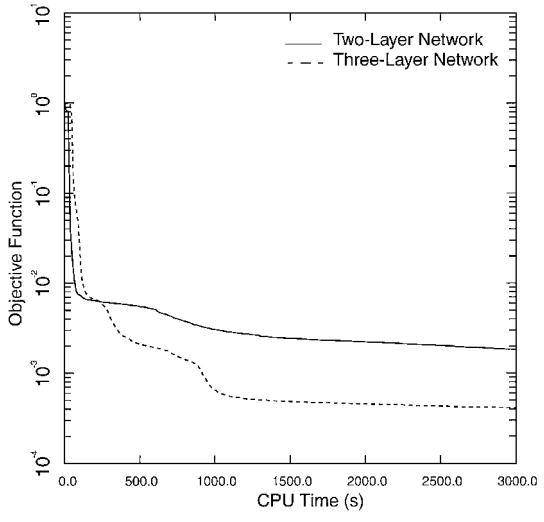

Fig. 15 Convergence characteristics of two- and three-layer neural networks.

to train the neural network becomes more critical. While simple techniques, such as judiciously representing each pressure distribution with a fewer number of points than that required for an accurate CFD solution, would reduce the CPU requirements for training, ultimately, it is imperative that rapid training methods be developed.

A preliminary evaluation of alternate neural network architectures has demonstrated that substantial reductions in the time required to train the network can be achieved. Figure 15 compares the training process for a two- and a three-layer feedforward neural network for a subset of the P&W design data. For this convergence study, the two-layer network had 41 neurons in the hidden layer. The three-layer network had 20 neurons in the first hidden layer and 7 in the second hidden layer. The number of neurons in the input and output layers was 5 and 1, respectively, for both cases. Note that the number of neurons was chosen so that the total number of weights in both cases was very nearly the same (246 in the two-layer case and 247 in the three-layer case). Figure 15 shows that for a given convergence criterion, such as reducing the value of the objective function by about three orders of magnitude, the three-layer network requires approximately one-fifth as much time to train as the two-layer network.

While this preliminary study shows that substantial reductions in training time can be achieved, more work is required to determine the optimal network architecture for a given problem. Such an optimal network architecture would not only result in rapid training, but would also require less simulation data than used in the current study to accurately represent the objective function in design space. Ideally, such a network would permit the design process to start from an initial airfoil geometry that is far from optimal. Progress toward such a design method is described in Ref. 15.

Conclusions

The focus of this research was on developing an intelligent design process keeping two specific objectives in mind. The first objective was to utilize the ability of the computer to interrogate and interpolate multidimensional unstructured data in designing turbomachinery, thus relieving the human designer to concentrate on higher-level decisions, such as recognizing geometric features that may enhance cooling, prevent separation, or delay transition. The second objective was to develop a design process that would permit the designer to rapidly perform a variety of tradeoff studies. Such a design process based on artificial neural networks is described in this paper. A preliminary evaluation of this design process has been conducted. The results are very encouraging and show that both these objectives have been met.

One of the most important advantages of the design process presented here is that once the neural network has been trained using a certain envelope of pressure distributions, new design targets that lie within this envelope can be obtained very rapidly. This advantage is not true of a brute-force-optimization technique because that technique resorts to function evaluations (CFD simulations) for every new design target. The ability to rapidly obtain new design targets is a major advantage because designers routinely perform a range of tradeoff studies (by modifying the design constraints) before arriving at the final design.

During this investigation it was found that training the neural network and generating the CFD data were straightforward tasks. Little or no fine-tuning of these procedures was required. Instead, much of the effort was expended on developing methods to generate complex airfoil shapes with as few geometric parameters as possible.

The process developed here is extendable to three-dimensional geometries. In fact, the neural network does not require any modification for a three-dimensional design. However, a three-dimensional design will increase the number of input parameters and will involve the use of larger amounts of training data. Hence, a network architecture that permits faster training, reduces the amount of simulation data required to accurately represent the objective function in design space, and allows the design to start from airfoil geometries that are far from optimal will help make the design process more efficient.

Acknowledgments

The authors would like to acknowledge the valuable technical discussions they have had with Frank Huber and Richard Rowey, formerly with Pratt and Whitney, West Palm Beach, Florida.

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]Rodi, P. E., "Three-Dimensional Hyperbolic Grid Generation Using Neural Network Controlled Governing Equations," AIAA Paper 96-0028, Jan. 1996.

[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, Aug. 1997.

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

[11]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.

[12]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.

[13]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.

[14]Dring, R. P., Joslyn, H. D., Hardin, L. W., and Wagner, J. H., "Turbine Rotor-Stator Interaction," *Journal of Engineering for Power*, Vol. 104, Oct. 1982, pp. 729–742.

[15]Rai, M. M., and Madavan, N. K., "Aerodynamic Design Using Neural Networks," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 173–182.