Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III) Editor: G.S. Dulikravich, Washington D.C. October 23-25, 199

DESIGN OF THRUST VECTORING EXHAUST NOZZLES FOR REAL-TIME APPLICATIONS USING NEURAL NETWORKS

Ravi K. Prasanth, Research Assistant
Robert E. Markin, Research Assistant
Kevin W. Whitaker, Assistant Professor
Aerospace Engineering Department
The University of Alabama
Tuscaloosa, AL 35487

ABSTRACT

Thrust vectoring is continuing to become an important issue in future military aircraft system designs. A recently developed concept of vectoring aircraft thrust makes use of flexible exhaust nozzles. Subtle modifications in the nozzle wall contours produce a non-uniform flow field containing a complex pattern of shock and expansion waves. The end result, due to the asymmetric velocity and pressure distributions, is vectored thrust. Specification of the nozzle contours required for a desired thrust vector angle (an inverse design problem) has been achieved with genetic algorithms. However, this approach is computationally intensive preventing nozzles from being designed in real-time which is necessary for an operational aircraft system. An investigation was conducted into using genetic algorithms to train a neural network in an attempt to obtain, in real-time, two-dimensional nozzle contours. Results show that genetic algorithm trained neural networks provide a viable, real-time alternative for designing thrust vectoring nozzles contours. Thrust vector angles up to 20° were obtained within an average error of 0.0914°. The error surfaces encountered were highly degenerate and thus the robustness of genetic algorithms was well suited for minimizing global errors.

INTRODUCTION

Future military aircraft will rely heavily on two- and three-dimensional thrust vectoring engines to boost their maneuverability and provide enhanced performance spanning their large operating envelopes. Current new technology engines use post-exit vanes or large moveable surfaces to redirect engine exhaust to yield the desired thrust vectoring. Although this method has proven to be effective, penalties must be paid. For example, most thrust vectoring devices are heavy, primarily due to structural requirements involving the impinging exhaust flow. The devices must also be designed to withstand the extreme temperatures of the engine exhaust gases impinging on them. Control of the vectoring apparatus is complex and adds even more weight to the aircraft. Furthermore, the installation of typical thrust vectoring devices tend to mandate large clearance gaps to allow surface movement and there is little opportunity for aerodynamic fairing. These and other factors can combine to yield higher overall drag forces on the aircraft.

A novel concept of vectoring engine thrust which addresses these concerns has been developed and shown to be viable [1]. The concept makes use of flexible nozzles where engine exhaust gases are turned not by some post-exit apparatus, but by *subtle* changes in the contour of the nozzle walls. The contour modifications produce a complex shock and expansion wave pattern in the nozzle flow field and the end result is vectored engine thrust. Through judicious tailoring of the nozzle contour, a large range of thrust vector angles may be achieved. Theoretical pitch vectoring of ±20° has already been demonstrated with this concept. Full three-dimensional vectoring (pitch, yaw, and roll) is currently being investigated and could possibly eliminate the need for any tail control surfaces on future aircraft. This would result in a tremendous savings in weight and drag as well as a significant reduction in radar cross-section.

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington, D.C., October 23-25, 199

This novel approach to thrust vectoring is based entirely on modifying the contour of the exhaust nozzle. In order for the technique to be useful in an operational aircraft system, the nozzle contour must be alterable in real-time. Structural concerns aside, the challenge is to specify, on demand, a nozzle contour for a pilot-requested thrust vector angle. This suggests that modification of the nozzle contour would be tied to the flight control system of the aircraft. What is necessary for the success of such a thrust vectoring system is the real-time solution of an inverse design problem. Simply stated, for a requested thrust vector angle, what would be the required nozzle contour?

Existing Jacobian-based methods for solving an inverse problem of this type are fraught with numerical difficulties and usually require an intense computational effort. A non-Jacobian based method like genetic algorithms can be used to compute the required nozzle contour for a requested thrust vector angle as was proven in a recent study by King, et al. [2]. However, the specification of the nozzle contours still could not be accomplished in real-time using genetic algorithms due to the computational requirements. Genetic algorithms, although they can routinely solve the inverse nozzle design problem (a definite advantage over many Jacobian based methods), still require numerous flow field evaluations to do so.

The hypothesis of the work presented here was that the inverse design problem could be solved in real-time if a non-Jacobian based method (genetic algorithms) was coupled with a neural network. Neural networks are biologically inspired computing systems with the phenomenal ability to grasp topological invariances that underlie inverse transformations. Thus, a neural network has the potential to be trained by a genetic algorithm and then, after sufficient training, would be able to solve the inverse nozzle design problem in real-time. It is important to note that there is no intention to dismiss Jacobian methods; in fact the coupling of a Jacobian method with a neural network to design nozzles is currently under investigation by the authors. In this paper, however, it is demonstrated that by using genetic algorithms, neural networks can be designed to provide an alternative with remarkable dexterity and computational ease for the real-time specification of thrust vectoring exhaust nozzles.

GENETIC ALGORITHM OVERVIEW

Genetic algorithms are increasing in popularity as a search and optimization technique but are still unknown to a large portion of the scientific community. Thus a brief description is in order. Genetic algorithms (GAs) are search algorithms based on the mechanics of genetics; they use operations found in natural genetics to guide their trek through a search space. Their main strength lies in their ability to perform efficiently across a broad spectrum of search problems including problems that are large, noisy, and poorly behaved. Two empirical investigations in the early 1970's demonstrated the technique's efficiency in function optimization [3, 4]. Subsequent application of GA's to the search problems of pipeline engineering, VLSI (very large scale integration) microchip layout, structural optimization, job-shop scheduling, medical image processing, propulsion system component design, and machine learning adds considerable evidence to the claim that GAs are broadly based and robust.

GAs consider many points in a search space simultaneously and therefore have a reduced chance of converging to a local optimum. In most conventional search techniques a single point is considered based on some decision rule. These methods can be dangerous in multi-modal (many peaked) search spaces because they can converge to local optima. However, GAs generate entire populations of points, test each point independently, and then combine qualities from existing points to form a new population containing improved points. Aside from conducting a more global search, the GA's simultaneous consideration of many points makes it highly adaptable to parallel processors since the evaluation of each point is an independent process.

GAs require the natural parameter set of the problem to be coded as a finite length string of characters. This is actually true of all operations performed on a computer at the machine

level, however the GA requires this coding on the local level. The user must represent possible solutions to the search problem as character strings. This may at first seem like an imposing task but there have been a number of techniques developed for coding solutions to search problems [5]. Since GAs work directly with a coding of the parameter set and not the parameters themselves, they are difficult to fool because they are not dependent upon continuity of the parameter space. A GA only requires information concerning the quality of the solution produced by each parameter set (objective function values). This differs from many optimization methods which require derivative information or, worse yet, complete knowledge of the problem structure and parameters. Since GAs do not require such problem-specific information they are more flexible than most search methods.

Lastly, GAs differ from a number of search techniques in that they use random choice to guide their search. Although chance is used to define their decision rules, GAs are by no means "random walks" through the search space. They use random choice efficiently in their exploitation of prior knowledge to rapidly locate optimal solutions.

NEUROMORPHIC APPROACHES TO INVERSE PROBLEMS

Before presenting the results of the neural network designed thrust vectoring nozzles, it is necessary to discuss the justification for solving an inverse problem using a non-Jacobian, genetic algorithm trained neural network approach. Of fundamental importance in solving inverse problems is the classic Stone-Weierstrass theorem [6, 7]. Using the Stone-Weierstrass theorem it can be shown that under certain conditions non-linear operators, such as the one encountered in fluid flow problems, can be represented using the well known Volterra and Wiener series thereby allowing computation of an approximate solution to the inverse problem. The impressive theoretical works of Volterra, Wiener and Urysohn (see Ref. [6]) on the characterization and approximation of non-linear operators find their full expression in neuromorphic approaches to inverse problem solving.

Let f and θ be Lebesgue integrable functions representing the spatio-temporal evolution of nozzle geometry and temporal evolution of thrust vector angle, respectively. The complex cause-effect structure that relates nozzle geometry and thrust vector angle can be written as

$$\theta = \mathbf{T}(f) \tag{1}$$

where $\mathbf{T} : \mathbf{E} \rightarrow \mathbf{F}$ is a mapping between appropriately defined Banach spaces $\mathbf{E}$ and $\mathbf{F}$. The inverse problem is to determine the map $\mathbf{T}^{-1} : \mathbf{F} \rightarrow \mathbf{E}$ such that

$$f = \mathbf{T}^{-1}(\theta) \tag{2}$$

Except in certain cases of little practical interest, the precise nature of the operator $\mathbf{T}$ is usually not known. Thus, to solve the inverse problem, we must first characterize the class of Banach space operators to which $\mathbf{T}$ belongs. But, even when $\mathbf{T}$ is known to belong to a certain class, $\mathbf{T}^{-1}$ may not exist as a unique map resulting in an infinity of solutions to the inverse problem. Therefore, we must approximate $\mathbf{T}^{-1}$ using fairly *nice* operators that lie *close* to $\mathbf{T}$ in some sense. Commonly used notions of *closeness* usually involve $\mathbf{L}^p$-norms defined on the terminal space $\mathbf{F}$:

$$L^2\ Norm: \quad \|\theta\|_2 = \int_0^T |\theta|^2 dt$$

$$L^\infty\ Norm: \quad \|\theta\|_\infty = ess\,\mathbf{sup}|\theta| \tag{3}$$

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S.Dulikravich, Washington D.C. October 23-25,1991

Several researchers have shown that infinite neural networks with a single hidden layer can approximate any Lebesgue measurable function [8, 9]. It has also been shown that L_2 (mean-square integrable) functions can be approximated by a three layer neural network [10]. These and other powerful results form the basis for applying neural networks to inverse problems. The canonical procedure for constructing a neuromorphic approximation to the inverse transformation is to capture topological invariances in the synaptic interconnections and weight structure using *a priori* generated training samples. Upon acquiring the invariances, a neural network can rapidly output a unique solution to any problem instance spanned by the training set.

To illustrate the advantages of a non-Jacobian method, consider a Jacobian based solution technique to a simple problem involving no unsteady effects. The goal is to find a static nozzle geometry f so as to minimize

$$\mathbf{J} = (\theta - \theta^*)^2 \tag{4}$$

subject to

$$\mathbf{T}f = \theta \tag{5}$$

where **J** represents the difference between the calculated and desired thrust vector angles. Under certain assumptions on **T**, variational calculus provides the necessary conditions for computing an *optimal* nozzle contour. In general, a numerical solution can then be found iteratively from

$$f_{new} = f_{old} + \mathbf{K}\ \nabla\mathbf{J} \tag{6}$$

where **K** is a gain and ∇**J** is the gradient of **J** evaluated at f_{old}. However, the disadvantages of this are:

1. Every time a thrust vector angle is demanded, the flow equations must be solved at every iteration until convergence in order to evaluate the gradient. This requires exceptional computing power for real-time applications.

2. The cost surface is highly degenerate and has a multitude of troughs. There is no guarantee that the iteration will converge to an acceptable solution.

3. Perhaps the most important limitation is that the *optimal* solution depends on the particular assumptions made regarding nozzle flow. The operator T that describes nozzle flow must be known explicitly for numerical implementation. Thus, experimental nozzle data cannot be used.

Consequently, the use of genetic algorithms for neural network design is justifiable. (As alluded to previously, a parallel research effort is currently underway at the University of Alabama to design a network using a Jacobian based back-propagation method and will be the subject of a future paper.)

NEURAL NETWORK DESIGN

Designing a feed forward neural network for real-time thrust vectoring involves two phases: a supervised training phase and a verification phase. Supervised training entails embedding the topological invariances in the synaptic weight space through repeated presentations of training samples that characterize the relationship between nozzle contour and thrust vector angle. Although a single network with a large number of synaptic interconnections can be designed to

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991

span the wide range of in-flight thrust vector angle requirements ($\pm 20°$), it is not ideally suited for real-time applications. Instead, designing several small neural networks with fewer real-time computations, each for a specified overlapping range of thrust vector angle, is more appropriate. Outputs from two neural networks that span the overlap containing the demanded thrust vector angle could be linearly interpolated to provide nozzle shapes. In addition to maintaining design simplicity, this approach has the significant advantage that the two neural networks can be run parallely, thereby reducing real-time computational requirements.

The feed forward neural network topology used in this study consists of a sigmoidal activation function, a single-node input layer, a four-node output layer, and two hidden layers each with four nodes. A schematic representation can be seen in Figure 1. Input to the neural network is the desired thrust vector angle θ; outputs from the network are polynomial coefficients $\{a_i,\ i = 1, 4\}$ that define the contour of the nozzle's upper wall as

$$f(x) = \sum_{i=1}^{4} a_i (x - x_0)^i (x - x_f)^i + g(x) \tag{7}$$

where x_0 and x_f are x-coordinates of the fixed ends of the baseline geometry $g(x)$. Only the upper nozzle wall was selected for modification to simplify this initial analysis. Thus, the neural network outputs define an incremental geometry referenced about the baseline.

The baseline nozzle developed for use in this study is shown in Figure 2. The nozzle type selected was a symmetric, dual expansion ramp nozzle with contourable walls. Concerns factored into the design were minimum length (to minimize weight) and reduced line of sight onto the engine hot-section to address observable characteristics. The baseline geometry was obtained after a number of iterations to insure the best performing nozzle was being used as a reference. The thrust vector angle of the baseline is zero degrees with a gross thrust coefficient of 0.983. In this study, a positive thrust vector angle corresponds to a vehicle nose-up pitching moment. It was assumed that the on-design conditions for the nozzle would be a nozzle pressure ratio of 10, a flight Mach number of 15, and a fluid specific heat ratio of 1.15. Being essentially a *proof-of-concept*, this study was also restricted to a two-dimensional (planar) nozzle to further simplify the analysis. However, except for an increase in the computational time required, no other technical challenges would be expected in the step from two to three dimensions.

Thrust vector angles corresponding to a large number of randomly generated, polynomial nozzle contours were computed using an analysis code based on the inverse method of characteristics [11]. This code, developed at the University of Alabama, allows for the analysis of supersonic flow fields internal to a nozzle as well as the supersonic exhaust plume. The code has been extensively validated with experimental data from NASA and industry. The network design procedure, however, does not depend on how the training samples are obtained and any method - numerical or experimental - can be used. Eight neural networks of identical topology were designed to span thrust vector angles between 5° and 20°. For each neural network design, 500 training samples that spanned the corresponding thrust vector angle range were presented to the network. Synaptic weights that minimized the ensemble error between neural network output $\{a_i,\ i = 1, 4\}$ and actual polynomial coefficients $\{c_i,\ i = 1, 4\}$ in the L^2 space were determined using a genetic algorithm. The set of weights displaying the minimum performance index over 25 generations was considered the optimal set of weights.

It must be noted that in an operational version of the neural network, inputs would be a function of time; whereas during the training phase, constant values of thrust vector angles constituted the training samples. This brings about a significant advantage of transforming what, in general, would be a dynamic optimization problem to a static network design problem. However, it is valid only upon neglecting unsteady fluid flow effects caused by dynamic changes in nozzle contour which is completely acceptable for aircraft thrust vectoring systems.

NEURAL NETWORK VERIFICATION

A Monte Carlo simulation was performed to verify the neural network design. 500 thrust vector angles between 5° and 20° were randomly generated. Each thrust vector angle was then presented to the neural network as an input. Polynomial coefficients obtained as outputs from the neural network were used to define a nozzle contour. Representative samples of nozzle contours obtained from the neural network can be seen in Figure 3. The MOC code was then run to find the actual thrust vector angle for each of the neural network specified contours. Figure 4 compares the requested thrust vector angle with the angle obtained from the MOC code. Figure 5 shows the error in the network achieved thrust vector angle. The performance of each of the eight networks used also can be clearly seen in Figure 5. Thrust vector angles of up to 20° were obtained within an average error of 0.0914° by affecting modifications to the upper nozzle wall only. Modifying both upper and lower walls would cause a very complex flow structure and could possibly expand the vectoring angle envelope. The maximum error in the thrust vector angle was 0.3791° which would be negligible in an operational aircraft system. Further improvements in vectoring performance can be expected to occur by using an L^{∞} type performance index and running the genetic algorithms in the training phase with an increased number of generations.

CONCLUSIONS

It has been have shown that neural networks provide a viable alternative to straight Jacobian based solution methods. They have significantly reduced real-time computations while maintaining accuracy and retaining design simplicity. In addition, although genetic algorithms may not be ideal for solving the inverse problem of thrust vectoring directly, their utility is demonstrated by their ability to train a neural network to do so.

The procedure presented here for designing neural networks and the subsequent design of thrust vectoring nozzles has advantages and disadvantages. Error surfaces encountered while designing thrust vectoring nozzles are highly degenerate and therefore a robust optimization scheme such as a genetic algorithm is required for global error minimization. But in problems of high dimensionality, there are numerical difficulties in using genetic algorithms, limiting the complexity of the simulated function and the size of the network that can be trained. Jacobian based back-propagation, for example, may reduce the training period significantly and would have no difficulty handling large dimensions. However, as with other gradient techniques, back-propagation is prone to converge to a local minimum, thereby converging to an incorrect network design or not converging at all. Further study is recommended to put these concerns to rest.

Finally, there are two competing aspects to neural network design - accuracy and generalizability - which need to be addressed. Accuracy has to do with how close is the approximation obtained using the neural network. Generalizability means that a neural network can interpolate and extrapolate beyond problem instances spanned by the training set. The performance indices used in this study do not reflect generalizability. It would be of considerable interest to develop performance indices that provide a balance between the two aspects and then redesign *adaptive* neural networks for thrust vectoring. In this study an off-line design method, wherein the entire training set is presented to the neural network at one time, was used; an on-line or adaptive neural network capable of learning while in operation would be better suited for practical applications.

REFERENCES

1. Whitaker, K. W., Gowadia, N. S., Fordyce, S. C., "Thrust Vectoring Using Non-Axisymmetric Nozzles With Flexible Contours," AIAA Paper No. 91-2368, 1991.

2. King, E. G., Freeman, L. M., Whitaker, K. W., and Karr, C. L., "Two-Dimensional Thrust Vectoring Nozzle Optimization Techniques," AIAA Paper No. 91-0473, 1991.

3. Holland, J. H., "Genetic Algorithms and the Optimal Allocations of Trials," *SIAM Journal of Computing*, Vol. 2, No. 2, p. 88, 1973.

4. Foo, N. Y. and Bosworth, J. L., "Algebraic, Geometric, and Stochastic Aspects of Genetic Operators," NASA CR-2099, 1972.

5. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.

6. Zimmer, R. J., "Essential Results of Functional Analysis," *Chicago Lectures In Mathematics*, The University Of Chicago Press, Chicago, 1990.

7. Royden, H. L., *Real Analysis*, MacMillan Publishing Co., New York, 1988.

8. Cotter, N. E., "Stone-Weierstrass Theorem and its Application To Neural Networks," *IEEE Transactions On Neural Networks*, Vol. 1, No. 4, December, 1990.

9. Gallant, A. R., White, H., "There Exists a Neural Network That Does Not Make Avoidable Mistakes," *Proceedings of IEEE International Conference On Neural Networks*, Vol. 1, 1988.

10. Hecht-Nielsen, R., *Neurocomputing*, Addison-Wesley Publishing Co., New York, 1990.

11. Whitaker, K. W. and Cates, J. E., "A User-Friendly Exhaust Nozzle Design Program Based on the Method of Characteristics," AIAA Paper No. 90-2029, 1990.

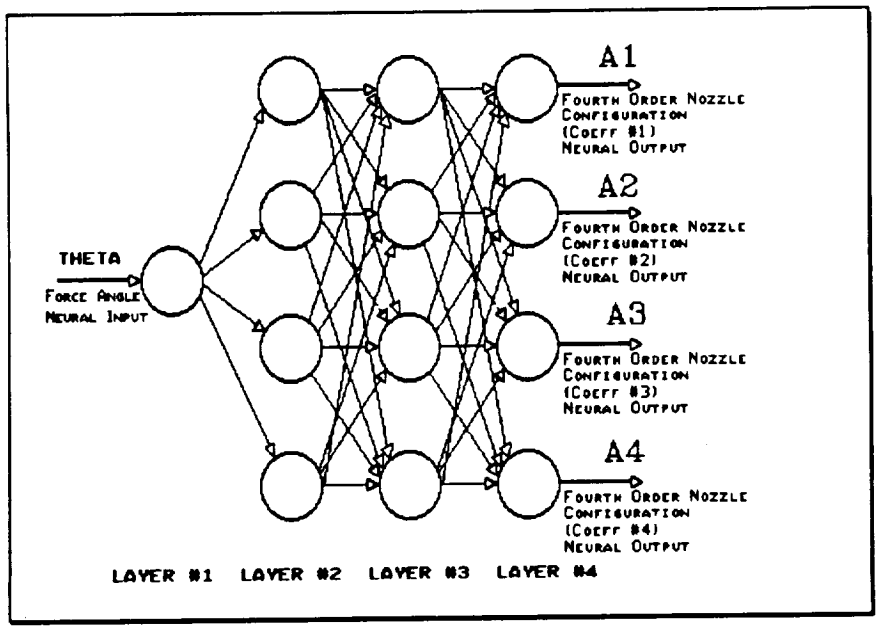

Figure 1. Schematic of Neural Network Used

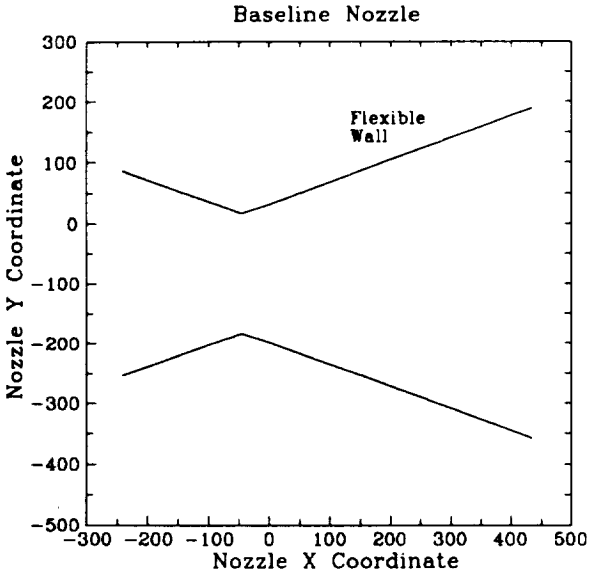

Figure 2. Baseline Nozzle Geometry

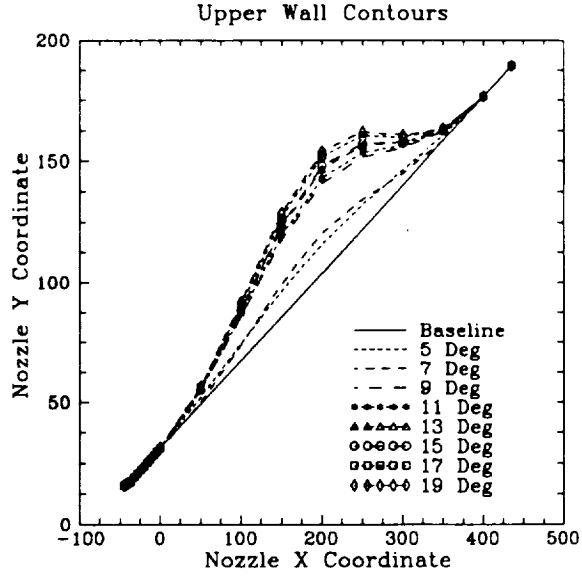

Figure 3. Representative Nozzle Contours Specified by Neural Network

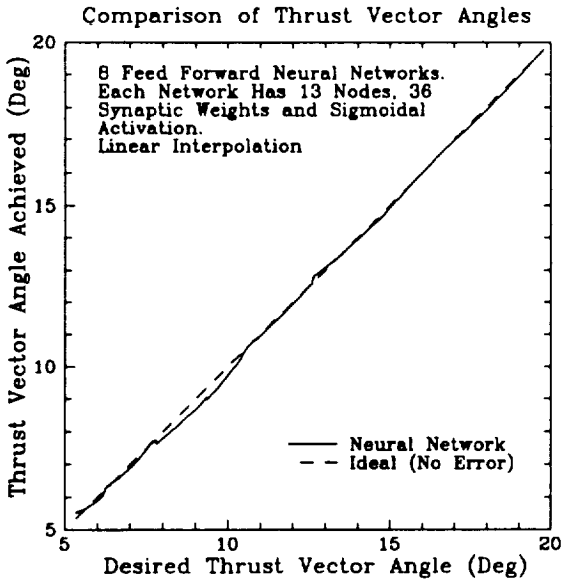

Figure 4. Comparison of Requested Angles with Angles Achieved

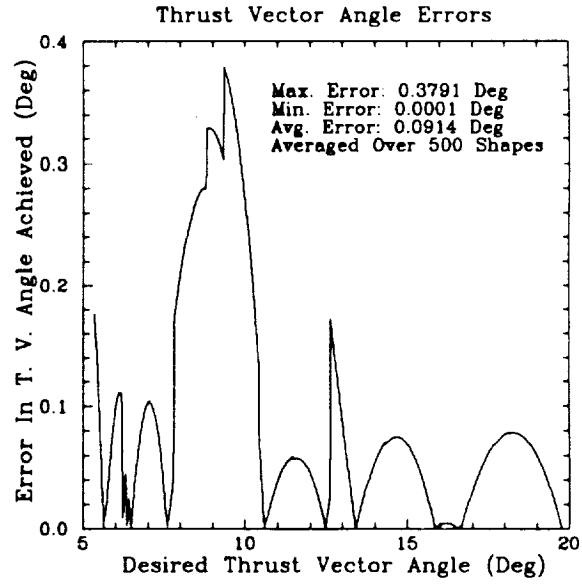

Figure 5. Error in Network Specified Angles