UNIVERSITY OF FLORIDA

Department of Aerospace Engineering,
Mechanics & Engineering Science

231 Aerospace Building
PO Box 116250
Gainesville, FL 32611-6250
Tel: (352)392-0961
Fax: (352) 392-7303

November 9, 2001

Mr. P. Kevin Tucker
NASA Marshall Space Flight Center
TD64 Applied Fluid Dynamics Group
Marshall Space Flight Center, AL 35812.

RE: NAG8-1742

Dear Mr. Tucker:

Please find enclosed the final report of the project entitled, "A Global Optimization Methodology for Rocket Propulsion Applications," which ended on September 15, 2001.

We have developed original techniques and applied them to address injector design optimization. In the process of developing the techniques, we have investigated ways to evaluate the performance of the optimization methodology, using analytical and empirical data, as well as CFD-based physical models. As evidenced by the report, several publications have been generated, summarized below.

Shyy, W., Papila, N., Vaidyanathan, R. and Tucker, P.K., "Global Design Optimization for Aerodynamics and Rocket Propulsion Components," Progress in Aerospace Sciences, Vol. 37, (2001) pp.59-118.

Shyy, W., Tucker, P.K. and Vaidyanathan, R., "Response Surface and Neural Network Techniques for Rocket Engine Injector Optimization," Journal of Propulsion and Power, Vol. 17, (2001) pp.391-401

Vaidyanathan, R., Papila, N., Shyy, W., Tucker, P.K., Griffin, L. W., Fitz-Coy, N. and Haftka, R.T, "Neural Network-Based and Response Surface-Based Optimization Strategies for Rocket Engine Component Design," 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Paper No. 2000-4480, Long Beach, CA, (2000).

Shyy, W., Papila, N., Tucker, P.K., Vaidynanthan, R., and Griffin, L. "Global Optimization for Fluid Machinery Applications," Keynote Paper, Proceedings of The Second International Symposium on Fluid Machinery and Fluid Engineering (ISFMFE), pp. 1-10, Beijing, China, October (2000).

Papila, N., Papila, M., Shyy, W., Haftka, R. and Fitz-Coy, N., "Accuracy Assessment of Response Surface Approximations" AIAA Paper # 2002-0539.

We greatly appreciate having this opportunity, and look forward to continuing our collaboration in the future. If you have any questions in regard to the final report, please let me know.

Sincerely,

Wei Shyy
Professor and Chairman

WS/jcm

cc: NASA Center for AeroSpace Information
NASA Grants Officer
Administrative Grants Officer

ACCURACY ASSESSMENT OF THE RESPONSE SURFACE METHOD

While the response surface method is an effective method in engineering optimization, its accuracy is often affected by the use of limited amount of data points for model construction. In this chapter, the issues related to the accuracy of the RS approximations and possible ways of improving the RS model using appropriate treatments, including the iteratively re-weighted least square (IRLS) technique and the radial-basis neural networks, are investigated. A main interest is to identify ways to offer added capabilities for the RS method to be able to at least selectively improve the accuracy in regions of importance. An example is to target the high efficiency region of a fluid machinery design space so that the predictive power of the RS can be maximized when it matters most. Analytical models based on polynomials, with controlled level of noise, are used to assess the performance of these techniques.

Scope

Specifically, the focus of this chapter is to address the following questions: (i) how to identify outliers associated with a given RS representation and improve the RS model via appropriate treatments? (i) how to focus on selected design data so that RS can give better performance in regions critical to design optimization? (iii) how to combine NN and polynomial techniques for improving the accuracy of the RS model? The following sections will give the details to each of these questions.

Iteratively Re-weighted Least Square (IRLS) Procedure

While constructing the RS, we often encounter the so-called *outliers*. Outliers are extreme cases on one variable, or a combination of variables, which have strong influences on the statistics and hence they should be carefully examined. These may reflect genuine properties of the underlying phenomenon, out of reach of a given polynomial-based RS, or be due to measurement errors or other anomalies, which should not be modeled [1, 2].

To highlight the impact of the outliers on RS accuracy, consider the case shown in Figure 1, where 20-data points, shown by the symbol ♦ in the figure, are supplied to fit an analytical function (Eqn. (3.10)). As shown, by including or excluding a single outlier is capable of considerably changing the slope of the regression line and, consequently, the accuracy of the approximation as shown in Figure 1. Of course, in many applications, it is not clear whether the disagreement between a RS and the training/testing data is caused by the outliers or the model accuracy. This is the main issue, which motivates our study here.

In this dissertation, *iteratively re-weighted least square* (IRLS) method originated by Beaton and Tukey [3] is applied to determine outliers. Iteratively re-weighted least square is a least square regression procedure where an additional scale factor (weight) is included in the fitting process and it can be adopted for detection of the outliers [4-7].

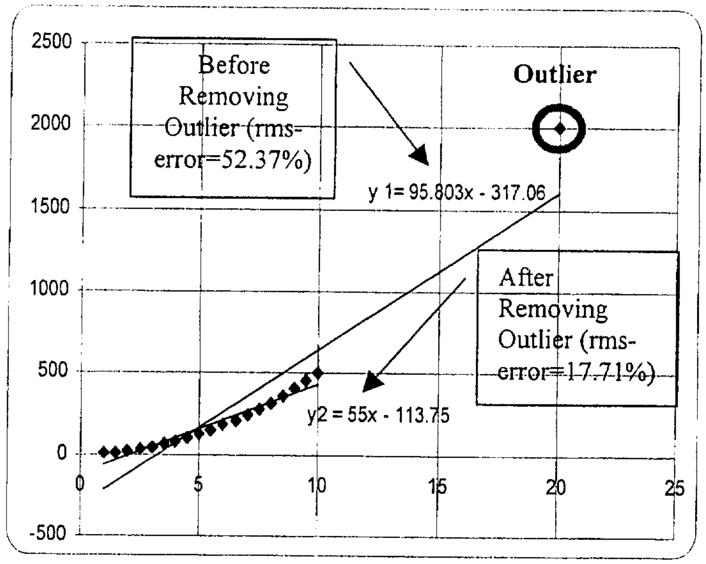


Figure 1: Effect of outliers on RS approximation and accuracy

The iteratively re-weighted least square procedure starts with an initial RS fitted to data and then low weights are assigned to points with large errors and the data refitted using a weighted least-square procedure. This process is repeated until convergence and weights of points that do not fit the underlying model (outliers) tend to converge small values. This effectively eliminates these points from the fitting process [4]. The weighting formula assigned to a data point is given by

$$w = \begin{cases} \left[1 - \left(\dfrac{\varepsilon/\sigma_a}{B}\right)^2\right]^2 & if\ \left|\varepsilon/\sigma_a\right| \le B \\ 0 & otherwise \end{cases} \tag{3.1}$$

where ε is the errors, σ_a is the *rms-error*, and B is a tuning constant [5]. Typically $1<B<3$, and closer to two, is suggested [6]. In this study, *1.9* is used for the tuning constant, B.

The coefficients of IRLS approximation are calculated by using the following formula

$$X^T W X b = X^T W y \tag{3.2}$$

where W is the diagonal weight matrix.

When calculating the coefficients in each iteration, the IRLS procedure employs the following step:

$$b^{(i+1)} = b^{(i)} + \left[X^T W^{(i)} X\right]^{-1} X^T W^{(i)} (y - X b^{(i)}) \tag{3.3}$$

and the initial coefficients are calculated by using

$$b = (X^T X)^{-1} X^T y \tag{3.4}$$

It should be noted that when all the weights are 1, the problem reduces to an ordinary least squares solution. In the present work, a configuration is determined by IRLS as *an outlier* if the weight is less than *0.01*.

By identifying the outliers in the RS construction, we can [4- 8]

- Gain better understanding of the scatter of the data generated by computational models or experimental measurements.
- Evaluate the effect of the outliers on the calculation of statistics and degree of fidelity of the RS model. The number of outliers can indicate the level of fidelity of the RS.
- Interpret and find ways to treat such design points without necessarily excluding them, because the so-called outliers can appear simply due to the lack of enough flexibility of a given polynomial used to construct the RS.

Biased Usage of Input Data

The outlier analyses can help assess the fidelity of each data point with a given RS, which, in turn, can aid selective refinement of the RS in regions of importance. Such approaches enable us to improve the model performance in critical areas and/or identify needs for further input data [9].

As already mentioned, in many design problems, we are more interested in data in certain regions. For example, in a fluid machinery design, the high efficiency region is of particular importance [10-11]. Within the limit of a quadratic or cubic polynomial-based RS, one may better achieve the optimization goal by placing higher emphases in these regions, even if this means that data in lower efficiency regions are fitted with worse accuracy than before. In the present approach, we identify data in selected regions and ensure that all of them receive emphasis, while employing the standard IRLS procedure to treat data in other regions. This procedure will be referred to here as the *biased IRLS* approach.

The main difference between the standard IRLS and the biased IRLS is the weight distribution used during IRLS procedure. In the standard IRLS procedure, the weight distribution given in Eqn. (3.1) is used and the model assigns the weights according to this formula. Whereas for the biased IRLS procedure, the weight distribution can be customized according to the region of interest. For example, we will consider $y \geq 0.5$ in Eqn. (3.10) as the region of interest.

For the biased IRLS procedure, weight distribution is chosen depending on the feasible region of the design problem. For example, Toropov and Alvarez [12] and van Keulen and Toropov [13] employed exponential weight distribution in order to improve computationally expensive and noisy response functions. The weight distribution used in this study is shown in Figure 2. Accordingly, the data with weights lower than w_{max}=*0.8*, due to the standard IRLS procedure, are artificially lifted to 0.8 if $y \geq y_{max} = 0.5$. To help devise a smooth weigh distribution, we have adopted the following steps: for those data points receiving the weight lower than w_{min}=*0.2* are forced to be 0.2 if $y \leq y_{min} = 0.2$; if $0.2 \geq y \geq 0.5$, then the following quadratic function is used to calculate the weight.

$$w = c_1 y^2 + c_2 y + c_3 \tag{3.5}$$

with

$$c_1 = \frac{w_{min} - w_{max}}{\left(y_{min} - y_{max}\right)^2} \tag{3.6}$$

$$c_2 = -2\frac{w_{min} - w_{max}}{\left(y_{min} - y_{max}\right)^2} y_{max} \tag{3.7}$$

$$c_3 = w_{max} + \frac{w_{min} - w_{max}}{\left(y_{min} - y_{max}\right)^2} y_{max}^2 \tag{3.8}$$

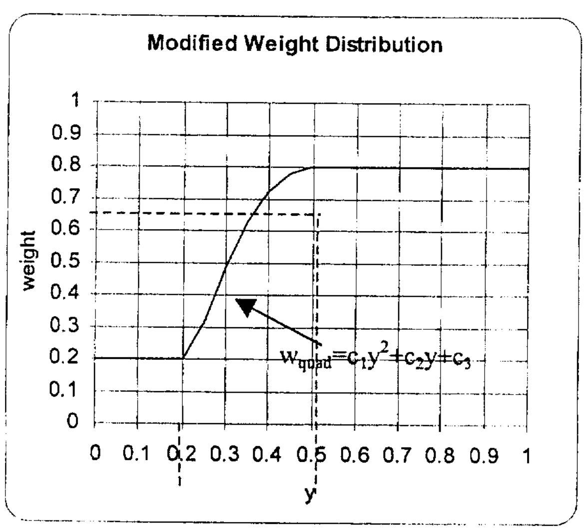


Figure 2: Modified weight distribution

Test Problems

An analytical model based on a polynomial is devised to help assess the performance of the various strategies. We investigate two different types of input data for constructing a RS: (1) data with no ambiguity, so that the RS model is expected to approximate the data as accurately as possible, and (2) data generated by adding noises generated by pseudo-random numbers to the analytical function, with the goal of evaluating the impact on RS's filtering capability.

Quartic Polynomial in 2-Dimension

A quartic function in 2-Dimension (2-D) is devised, as shown in Eqn. (3.10).

$$y_1 = 2000 + 1000x_1^3x_2 - 500x_2^2 + 25x_1^3 + x_1^2x_2 + x_1^2 + x_1^3 + x_1x_2^2 + x_1x_2 + x_2^2 \tag{3.9}$$

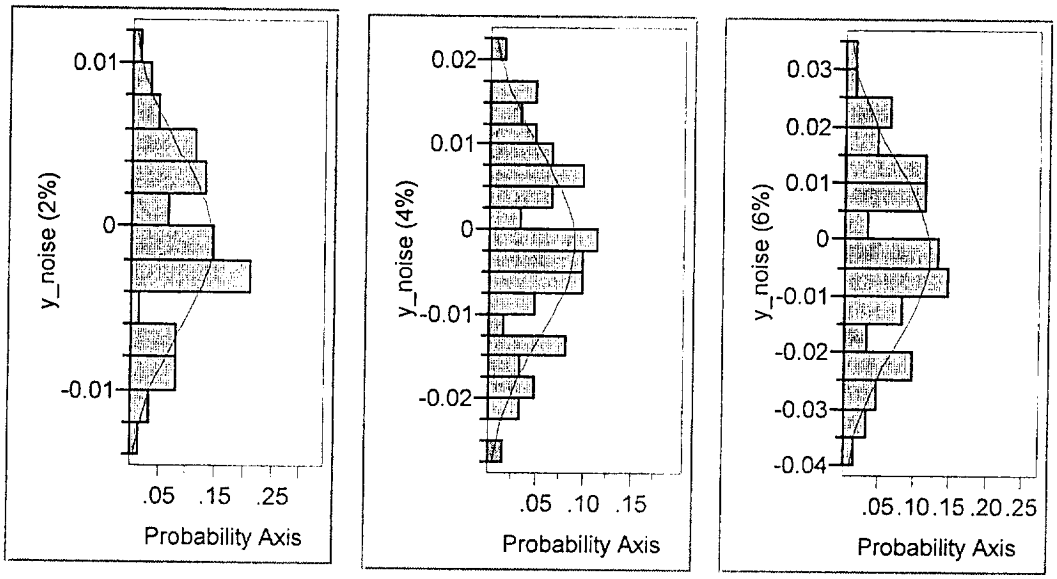
Equation (3.9) can be normalize into (0,1) region using

$$y = \frac{y_1 - min(y_1)}{max(y_1) - min(y_1)} \tag{3.10}$$

Using Eqn. (3.10), 61 data are generated. Nine of them are selected based on *FCCD* and the rest of 52 designs are selected based on *OA* design. 465-additional data is generated for testing.

Quartic Polynomial with Noise

The normally distributed pseudo-random data is added to Eqn. (3.10) to examine the effect of noise on RS approximation accuracy. Noise distribution of 2% (Figure 3a), 4% (Figure 3b), and ±6% (Figure 3c) are considered to see the effect of amount of noise.



(a) 2% Noise (b) 4% Noise (c) 6% Noise

Figure 3: Noise distributions to the original analytic data (# of data points: 465)

Results

We have studied three quadratic approximation models:

- RS without outliers treatment,
- Standard IRLS
- Biased IRLS

A framework combining NN and polynomial RS techniques is also employed in order to identify additional ways to improve the accuracy of the RS model. The results of these three quadratic approximation models and NN-enhanced RSs, for the quartic equation-generated data, with and without noise, are summarized in the following sections.

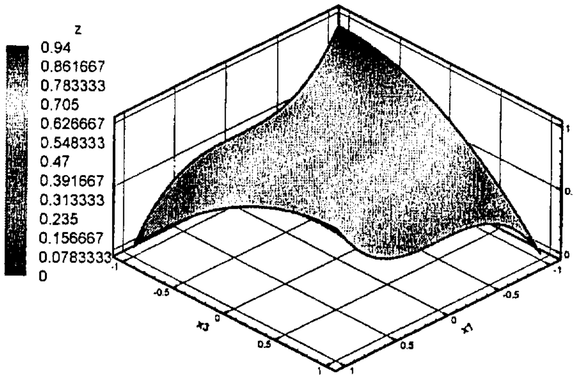
Polynomial-based RSs

The response surface generated by considering all input data with equal weighting, along with the standard IRLS and biased IRLS models, obtained for the no noise case are illustrated in Figure 4.

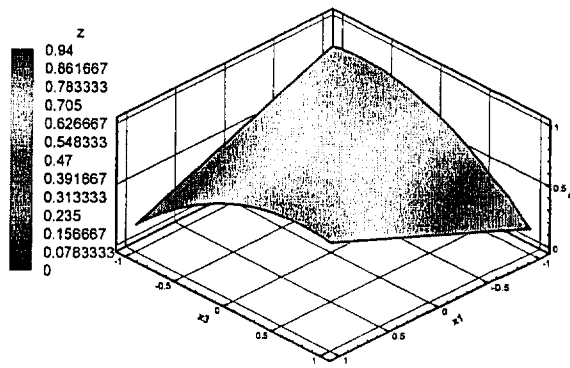
The standard IRLS procedure identifies four outliers for data with 2%, 4% and 6% noise and without noise. Out of the four, two of them are in the $y \geq 0.5$ region. The biased IRLS procedure detects two outliers for all cases. None of them is in $y \geq 0.5$ by design.

The statistical summaries of the RS, the standard IRLS and the biased IRLS models constructed for data, with and without noise, are shown in Table 1. Without the influence of noise, even though the overall testing error of the biased IRLS model is higher than the original RS, it is most accurate in $y \geq 0.5$ (Figure 5). The same observation holds for 2%, 4%, and 6% noise cases.

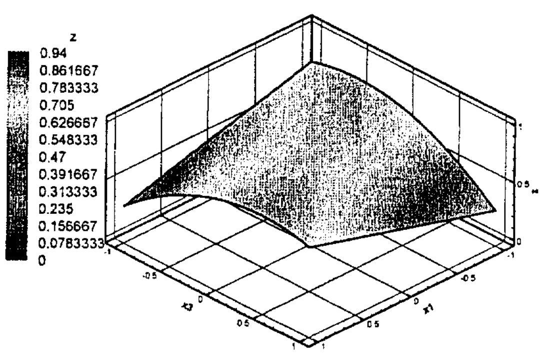
It appears that while noises can affect the performance of the RS, one can consistently target the region of importance and improve the accuracy of a RS there.



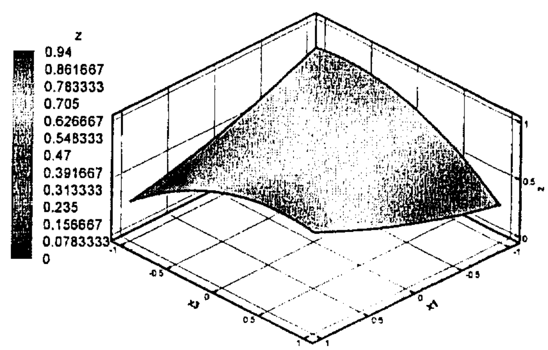
(a) Analytic function



(b) Quadratic RS



(c) Standard IRLS

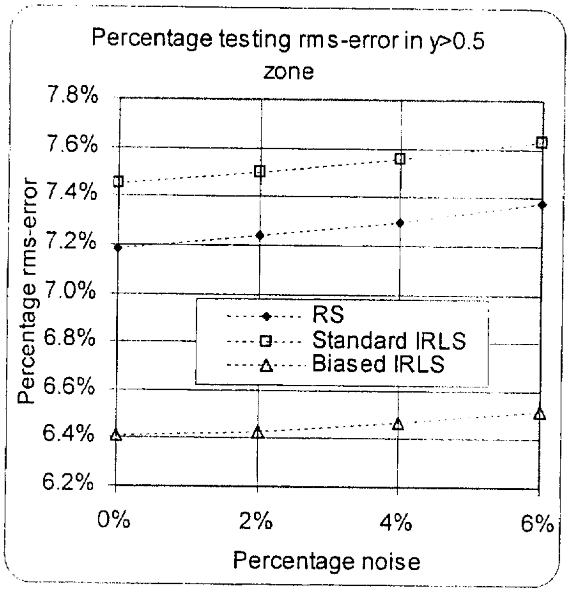


(d) Biased IRLS

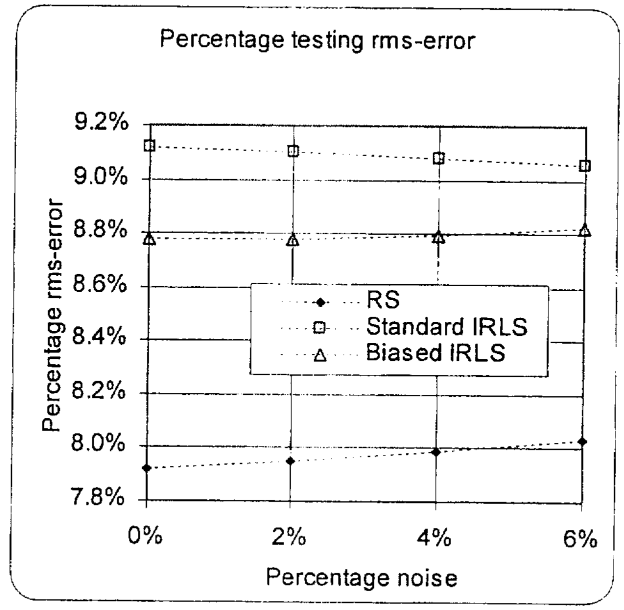
Figure 4: Quartic function, RS, standard IRLS, and biased IRLS approximations for analytical data without noise

Table 1: Statistical summaries of different quadratic models (a) original RS, (b) standard IRLS, and (c) biased IRLS for analytical data with no noise, 2%, 4%, and 6% noise (# of training data=61, # of test data=465, and # of test data in y>0.5=405)

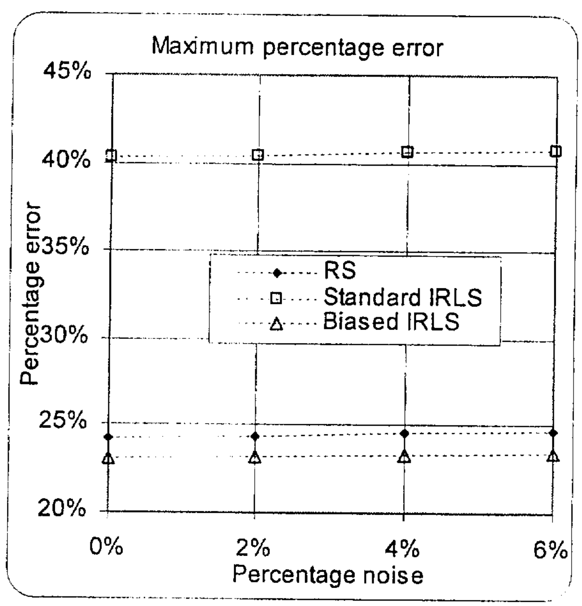
	RS	**Standard IRLS**	**Biased IRLS**
RSquare	0.89	0.84	0.85
RSquare Adj	0.88	0.82	0.84
%Training rms-Error	9.22%	11.36%	10.63%
Mean of Response	0.64	0.64	0.65
% Testing rms- Error	7.92%	9.12%	8.78%
% Testing rms-Error in y>0.5	7.18%	7.45%	6.41%
# of Training Data in y>0.5	51	49	51
Max Error	24.24%	40.28%	23.15%
2% NOISE			
RSquare	0.89	0.83	0.85
RSquare Adj	0.88	0.82	0.83
%Training rms-Error	9.41%	11.52%	10.82%
Mean of Response	0.64	0.64	0.65
% Testing rms- Error	7.95%	9.10%	8.78%
% Testing rms-Error in y>0.5	7.24%	7.50%	6.43%
# of Training Data in y>0.5	52	50	52
Max Error	24.40%	40.44%	23.26%
4% NOISE			
RSquare	0.88	0.83	0.84
RSquare Adj	0.87	0.81	0.83
%Training rms-Error	9.67%	11.74%	11.10%
Mean of Response	0.64	0.64	0.65
% Testing rms- Error	7.99%	9.08%	8.79%
% Testing rms-Error in y>0.5	7.30%	7.56%	6.47%
# of Training Data in y>0.5	52	50	52
Max Error	24.56%	40.60%	23.34%
6% NOISE			
RSquare	0.88	0.82	0.83
RSquare Adj	0.87	0.81	0.82
%Training rms-Error	10.00%	12.01%	11.43%
Mean of Response	0.64	0.64	0.65
% Testing rms- Error	8.03%	9.06%	8.82%
% Testing rms-Error in y>0.5	7.38%	7.63%	6.52%
# of Training Data in y>0.5	53	51	53
Max Error	24.72%	40.73%	23.43%



(a) % Testing error in y>0.5



(b) % Testing error



(c) % Maximum error

Figure 5: Percentage error versus noise distributions for original RS, standard IRLS, and biased IRLS

Neural Network Training

Three different RBNN designs are considered: *solveRB*, and two RBNN designs created by Orr [14-17] by using ridge regression (RRNN) and regression trees (RTNN) methods. The relative performances of these designs are compared and NN-enhanced RSs are constructed by using each of these designs.

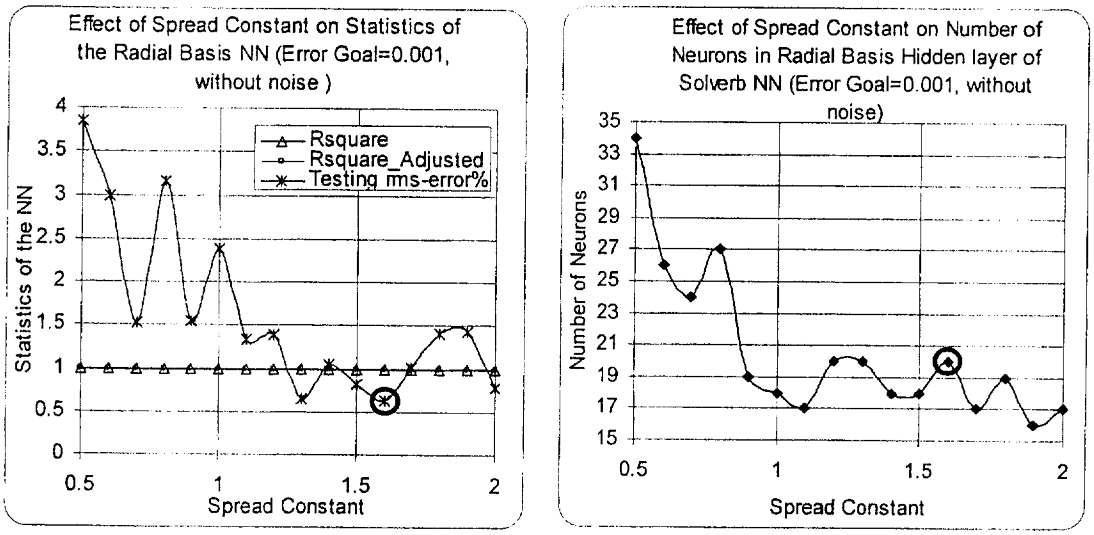
While training RBNNs, the first step is to decide test/train partitions of the available data. For this purpose, a *5-fold cross-validation* technique is used. The results of 5-fold cross-validation for 61-data obtained for the analytical function, given by Eqn. (3.1), are shown in Table 2.

Table 2: Results of 6-fold cross validation for analytical data without noise (# of training data=51 and # of testing data=10)

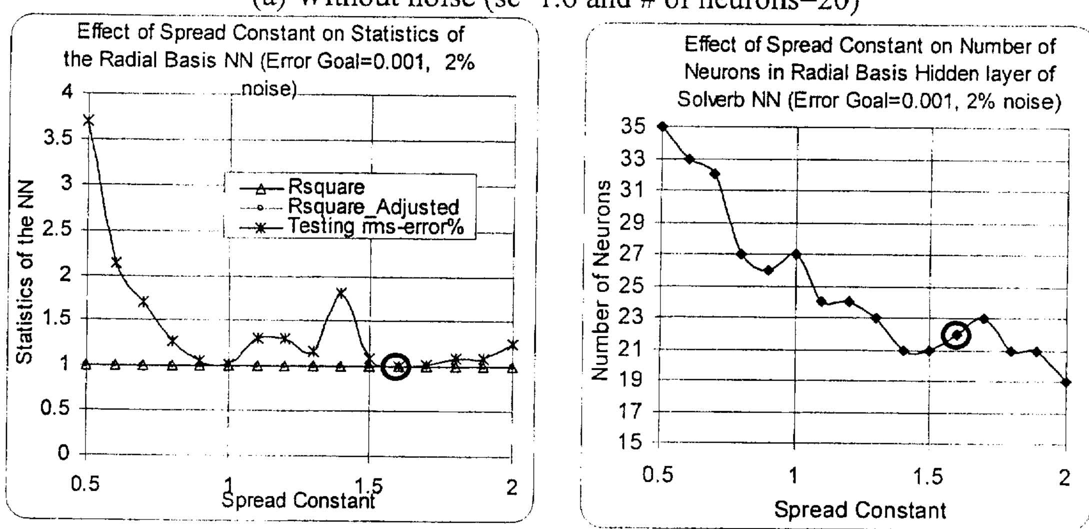
	RSquare	RSquare Adj	Training rms-error %	Mean of Response	Testing rms-error %
Data Set #1	0.89	0.88	9.48%	0.64	9.88%
Data Set #2	0.90	0.89	9.41%	0.64	9.66%
Data Set #3	0.88	0.87	10.61%	0.63	9.80%
Data Set #4	**0.90**	**0.89**	**9.50%**	**0.63**	**9.42%**
Data Set #5	0.90	0.89	9.84%	0.63	8.17%
Average	**0.90**	**0.88**	**9.77%**	**0.64**	**9.39%**

Accordingly, *data set #4* which has the closest statistics to the average statistics is chosen as the test/train partition of the NN training.

While training *solveRB*, two network design parameters need to be optimized: the spread constant, *sc*, and the error goal. The *solveRB* networks are trained by using the range of *sc* and error goal in order to select the optimum values for these parameters yielding the smallest testing *rms-error* of the network shown in Figure 6 and Figure 7. The same ideas hold for Orr's RRNN and RTNNs.

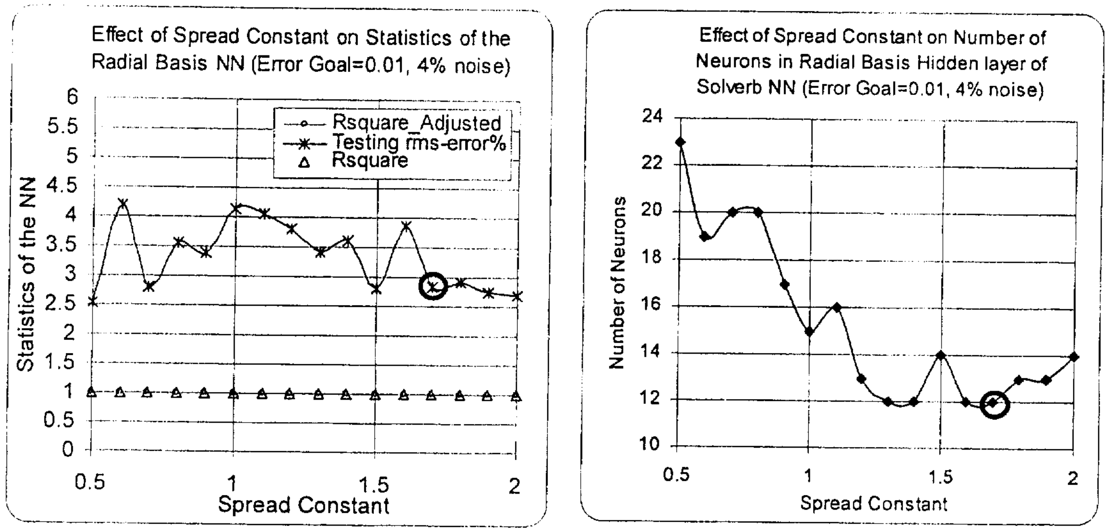


(a) Without noise (sc=1.6 and # of neurons=20)

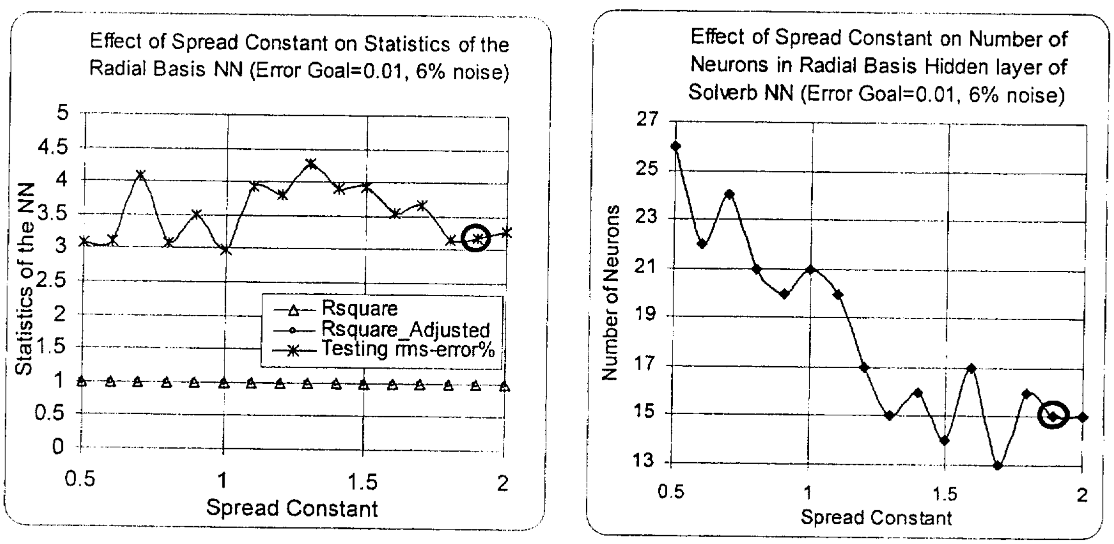


(b) 2% noise (sc=1.6 and # of neurons=22)

Figure 6: Optimization of *solveRB* network design parameters, *sc,* for analytical data with no noise and 2% noise (error goal=0.001)



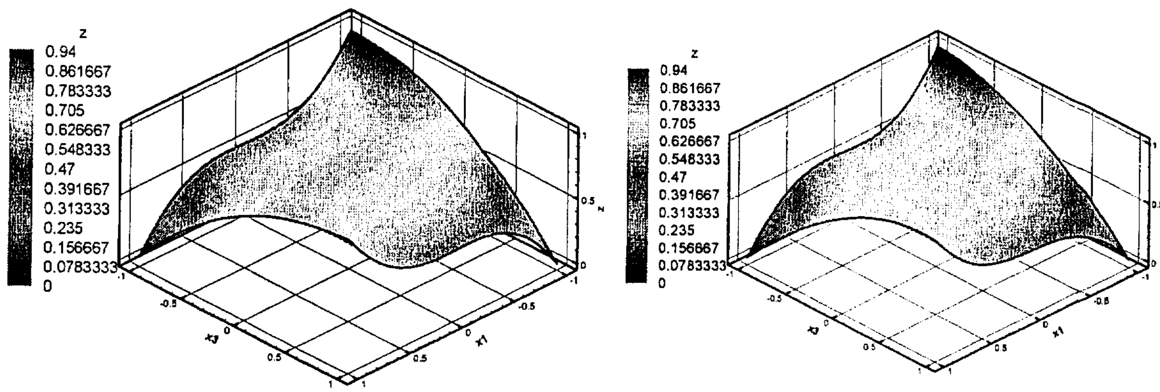
(a) Without noise (sc=1.7 and # of neurons=12)



(b) 2% noise (sc=1.9 and # of neurons=15)

Figure 7: Optimization of *solveRB* network design parameters, *sc,* for analytical data with 4% and 6% noise (error goal=0.01)

Using the test/train (51/10) partition of the *data set # 4*, three RBNNs are trained. Figure 8 shows the *solveRB* approximations for the data set without noise and with the highest noise level of 6%. Both figures agree well with the true function shown in Figure 4 (a).



(a) No noise (b) 6% Noise

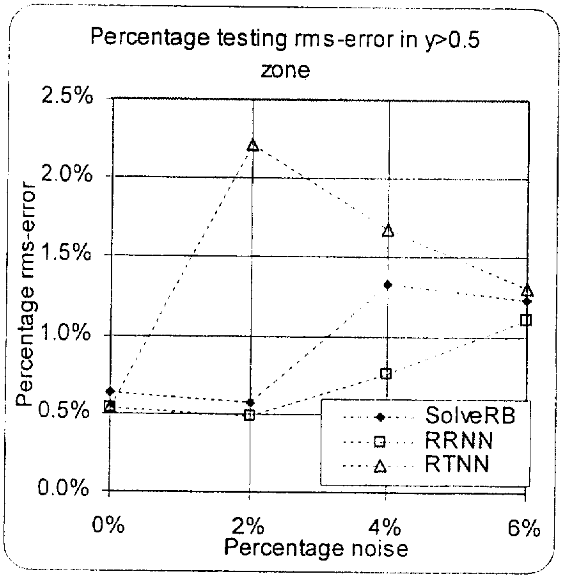
Figure 8: *SolveRB* approximation for analytical data without noise and with 6% noise

Table 3 shows the statistics of the approximations obtained by using *solveRB* NNs, RRNN and RTNN for data sets with 2%, 4%, and 6% noises, and without noise. Besides testing error obtained by using the10-test data as a part of NN training, a larger data set with 465 data is also used to test the prediction accuracy of the NN approximation. All of these results are presented in Table 3.

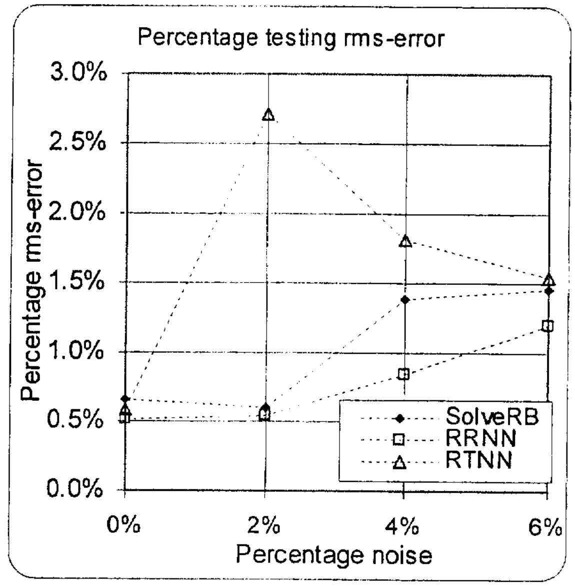
According to this table, all three RBNN methods for the data without noise can be accurately trained with testing accuracy of better than 1% in the entire domain. This is still the case for *solveRB* and RRNN when the noise level is 2%. As the noise level increases, the overall error and error in the region of interest increases but remains lower than 2%. Among the three alternatives of RBNN designs, *solveRB* and RRNN has comparable performance whereas RTNN is less accurate (Figure 9).

Table 3: Statistical summaries of RBNN training for analytical data with no noise, 2%, 4%, and 6% noise (# of training data=51, # of test data=465, and # of test data in y>0.5=405)

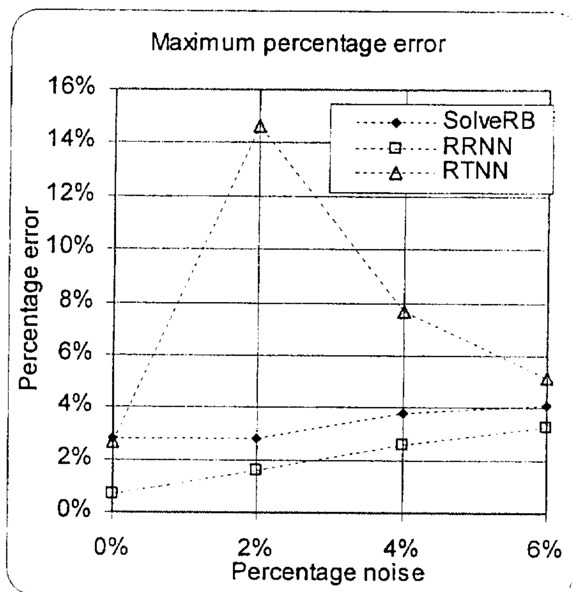
	SolveRB	RRNN	RTNN
RSquare	1.00	1.00	1.00
RSquare Adj	0.99	0.99	1.00
Mean of Response	0.68	0.68	0.67
%Testing rms-Error with 10 Data as a part of Training	0.94%	1.07%	0.77%
% Testing rms-Error	0.66%	0.52%	0.58%
% Testing rms-Error in y>0.5	0.64%	0.53%	0.53%
# of Training Data in y>0.5	42	42	42
Max Error	2.82%	0.71%	2.67%
2% NOISE			
RSquare	0.99	0.99	0.98
RSquare Adj	0.98	0.98	0.96
%Testing rms-Error with 10 Noisy Data as a part of Training	1.48%	1.78%	2.24%
Mean of Response	0.68	0.68	0.68
% Testing rms-Error	0.60%	0.54%	2.72%
% Testing rms-Error in y>0.5	0.57%	0.48%	2.22%
# of Training Data in y>0.5	43	43	43
Max Error	2.79%	1.60%	14.62%
4% NOISE			
RSquare	0.94	0.96	0.94
RSquare Adj	0.87	0.92	0.86
%Testing rms-Error with 10 Noisy Data as a part of Training	4.08%	3.16%	4.14%
Mean of Response	0.67	0.68	0.68
% Testing rms-Error	1.38%	0.84%	1.82%
% Testing rms-Error in y>0.5	1.33%	0.76%	1.68%
# of Training Data in y>0.5	43	43	43
Max Error	3.76%	2.58%	7.66%
6% NOISE			
RSquare	0.92	0.92	0.92
RSquare Adj	0.82	0.82	0.81
%Testing rms-Error with 10 Noisy Data as a part of Training	4.70%	4.75%	4.80%
Mean of Response	0.68	0.68	0.67
% Testing rms-Error	1.46%	1.20%	1.55%
% Testing rms-Error in y>0.5	1.23%	1.11%	1.31%
# of Training Data in y>0.5	43	43	43
Max Error	4.07%	3.26%	5.14%



(a) % Testing error in y>0.5



(b) % Testing error



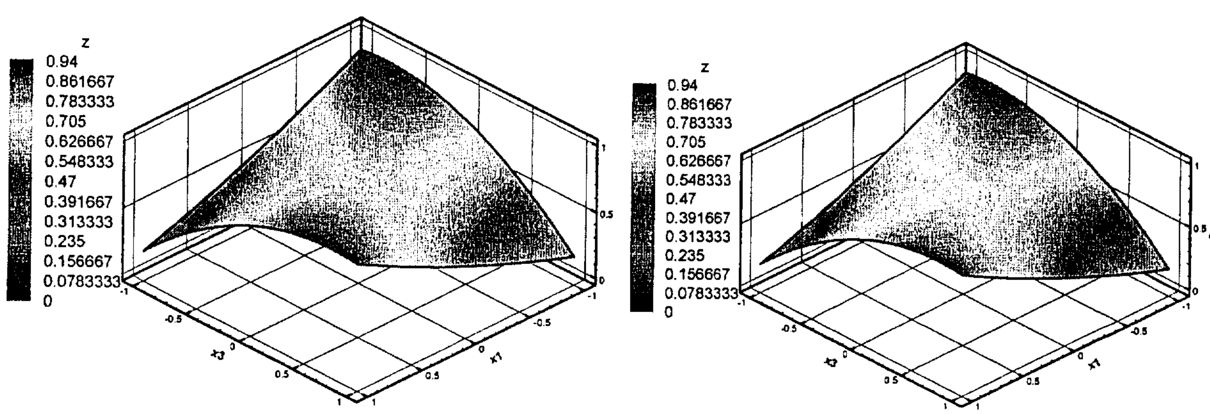
(c) % Maximum error

Figure 9: Percentage error versus noise distributions for *solveRB*, RRNN, and RTNN approximations

Neural Network-Enhanced RS

To investigate whether on can use the NN-predicted data to help improve the predictive capability of the polynomial-based RSM, a subset of 25 data (??) is selected from the entire set of 465 data. These 25 data points correspond to the highest y among all 465 data, and are added to the original training set of 61 data to enrich the data concentration in the region of interest.

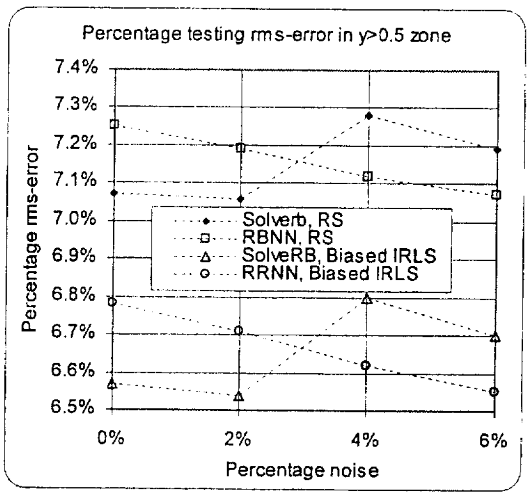
The performance of the *solveRB*-enhanced RS, with and without noise, is demonstrated Figure 10.



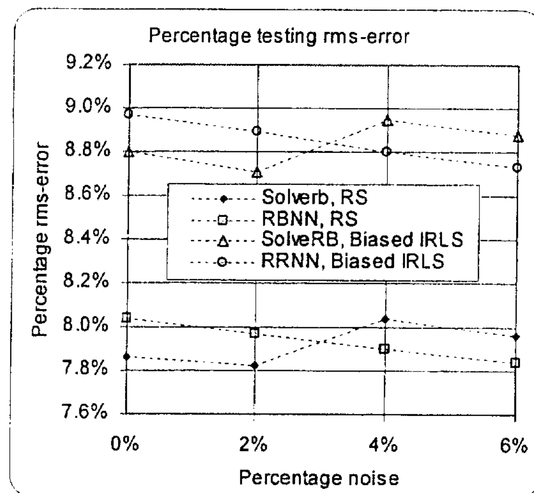
(a) No noise (b) 6% noise

Figure 10: *SolveRB*-enhanced quadratic RS for analytical data with and without noises.

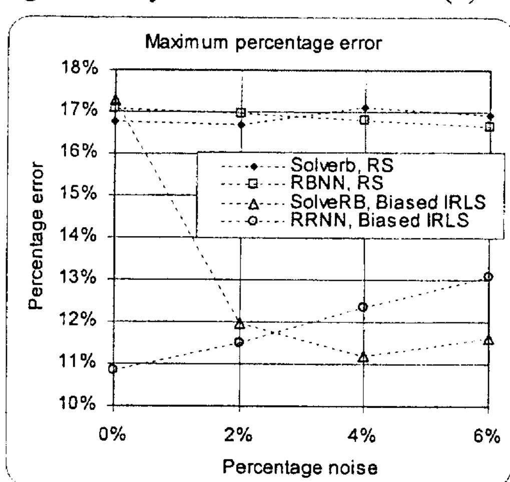
Table 4 compares the statistics of the RRNN-enhanced and *solveRB*-enhanced RS, both with the biased IRLS procedure. From this table, it is observed that RBNN can be effectively combined with the biased IRLS procedure, yielding better accuracy in the targeted region, i.e., $y \geq 0.5$, in all cases. The opposite is true in terms of the overall testing accuracy. It is also noted that the testing accuracy of *solveRB*-enhanced models are better than RRNN-enhanced models up to 2% noise. As the noise increases to 4% and beyond, the RRNN-enhanced models outperform the *solveRB*-enhanced models (Figure 11). By comparing Table 4 with Table 1, it can be seen that while the NN-enhanced RS models give better training statistics than the original RS, it doesn't seem to improve the effectiveness of the biased-IRLS treatment with no enhancement from the NN at all. This outcome is somewhat surprising because substantially more data in the targeted region are used to construct the RS.



(a) % Testing error in y>0.5



(b) % Testing error



(c) % Maximum error

Figure 11: Percentage error versus noise distributions for *solveRB*-, RRNN-, and RTNN-enhanced RS and biased IRLS approximations

Table 4: Comparison of statistical summaries of RBNN-enhanced RS and RBNN-enhanced biased IRLS for analytical data with no noise, 2%, 4% and 6% noise (# of training data=86, # of training data in y>0.5=76, # of test data=440, and # of test data in y>0.5=380)

	SolveRB-Enhanced RS	**RRNN-Enhanced RS**	**SolveRB-Enhanced Biased IRLS**	**RRNN-Enhanced Biased IRLS**
RSquare	0.92	0.92	0.90	0.90
RSquare Adj	0.92	0.92	0.89	0.89
%Training rms-Error	7.54%	7.70%	8.58%	8.72%
Mean of Response	0.72	0.72	0.72	0.73
% Testing rms-Error	7.86%	8.04%	8.80%	8.97%
% Testing rms-Error in y>0.5	7.07%	7.25%	6.57%	6.78%
Max Error	16.75%	17.08%	12.60%	10.84%
2% NOISE				
RSquare	0.92	0.92	0.90	0.90
RSquare Adj	0.92	0.92	0.89	0.89
%Training rms-Error	7.56%	7.67%	8.58%	8.69%
Mean of Response	0.72	0.72	0.72	0.72
% Testing rms-Error	7.82%	7.97%	8.71%	8.89%
% Testing rms-Error in y>0.5	7.06%	7.19%	6.54%	6.71%
Max Error	16.71%	16.95%	11.97%	11.49%
4% NOISE				
RSquare	0.92	0.92	0.90	0.90
RSquare Adj	0.92	0.92	0.89	0.89
%Training rms-Error	7.75%	7.64%	8.75%	8.66%
Mean of Response	0.72	0.72	0.73	0.72
% Testing rms-Error	8.04%	7.90%	8.95%	8.80%
% Testing rms-Error in y>0.5	7.28%	7.12%	6.80%	6.62%
Max Error	17.12%	16.79%	11.21%	12.34%
6% NOISE				
RSquare	0.92	0.92	0.90	0.90
RSquare Adj	0.92	0.92	0.89	0.89
%Training rms-Error	7.68%	7.61%	8.69%	8.63%
Mean of Response	0.72	0.71	0.72	0.72
% Testing rms-Error	7.96%	7.84%	8.88%	8.73%
% Testing rms-Error in y>0.5	7.19%	7.07%	6.70%	6.55%
Max Error	16.93%	16.64%	11.61%	13.08%

Summary

In this chapter, standard IRLS, biased IRLS method placing higher emphasis on data belonging to a region of interests and a global optimization framework combining NN and polynomial RS techniques are used as searching for possible ways of improving the RS model. Analytical models based on a cubic polynomial in 7-D, with and without noise, are used to assess the effectiveness of these techniques. The results obtained can be summarized as follows:

- High-amplitude noise designs (outliers) are identified by IRLS methods. Since the true function is known to be cubic polynomial, the reason of the designs determined as outliers in this paper is modeling inaccuracy.
- Accuracy of the RS can be enhanced in regions of interest by increasing the weighting of the data in that region while constructing the RS. Such a method is found to be useful when modeling error dominates the noise.
- Neural networks are effective tools for approximation as long as they are well trained. One of the critical issues when training the NN accurately is to decide on the test/train partitions of the data. Cross-validation method can be used for the purpose.
- The NN-enhanced RSM can be quite useful depending on how well and accurate the NN is trained. NN-enhanced RSM may reduce the required number of computational calculations required to construct an accurate RS substantially. However, the performance of such an approach can be affected by the amount of the noise of the computational/experimental data.

References

1. *StatSoft, Inc. Electronic Statistics Textbook*, Tulsa, OK, 2001, WEB: http://www.statsoft.com/textbook/stathome.html.

2. Myers, R. H., and Montgomery, D. C., *Response Surface Methodology –Process and Product Optimization Using Designed Experiments*, New York: John Wiley & Sons, Inc., 1995.

3. Beaton, A. E., and Tukey, J. W., "The fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data," *Technometrics*, Vol. 16, No.2, 1974, pp.147-185.

4. Papila, M. and Haftka, R. T., "Response Surface Approximations: Noise, Error Repair and Modeling Errors," *AIAA Journal*, Vol.38, No.12, 2000, pp. 2336–2343.

5. Holland, P. W., and Welsh, R. E., "Robust Regression using Iteratively Reweighed Least-Squares," *Communications Statistics: Theory and Methods*, A6 (9), 1977, pp.813-827.

6. Myers, R. H., "*Classical and Modern Regression with Applications*, " Boston:PWS-KENT, 1989.

7. Kim, H., Papila, M., Mason W. H., Haftka, R. T., Watson, L. T., and Grossman, B., "Detection And Correction Of Poorly Converged Optimizations By Iteratively Reweighted Least Squares," 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, Atlanta, Georgia, AIAA 2000-1525, 2000.

8. Papila, M. and Haftka, R. T., "Uncertainty and Response Surface Approximations", 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Seattle, Washington, AIAA 2001-1680, 2001.

9. Balabanov, V.O., Giunta, A.A., Golovidov, O., Grossman, B., Mason, W.H., Watson, L.T., and Haftka, R.T., "Reasonable Design Space Approach to Response Surface Approximation," *Journal of Aircraft,* Vol. 36, No.1, 1999, pp. 308-315.

10. Papila, N., Shyy, W., Griffin, L., Huber, F. and Tran, K., "Preliminary Design Optimization for a Supersonic Turbine for Rocket Propulsion," AIAA/SAE/ASME/ASEE 35th Joint Propulsion Conference, AIAA 2000-3242, 2000.

11. Shyy, W., Papila, N., Vaidyanathan, R. and Tucker, P.K., "Global Design Optimization for Aerodynamics and Rocket Propulsion Components, " *Progress in Aerospace Sciences,* Vol. 37, 2001, pp.59-118.

12. Toropov, V.V.and Alvarez, L.F., "Approximation model building for design optimization using genetic programming methodology," 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, AIAA 1998-4769, 1998.

13. van Keulen, F. and Toropov, V.V., "New developments in structural optimization using adaptive mesh refinement and multi-point approximations," *Engineering Optimization*, vol. 29, 1997, pp. 217-234.

14. Orr, M. J. L., "Introduction to Radial Basis Function Networks," Centre for Cognitive Science, Edinburgh University, EH 9LW, Scothland, UK, 1996, WEB: http://www.anc.ed.ac.uk/~mjo/rbf.html.

15. Orr, M. J. L., "Optimising the Widths of RBFs," Fifth Brazilian Symposium on Neural Networks, Belo Horizonte, Brazil, 1998

16. Orr, M. J. L., "Recent Advances in Radial Basis Function Networks," Technical Report, Institute for Adaptive and Neural Computation Division for Informatics, Edinburgh University, EH 9LW, Scotland, UK, 1999, WEB: http://www.anc.ed.ac.uk/~mjo/rbf.html.

17. Orr, M. J. L., "Matlab Functions for Radial Basis Function Networks," Institute for Adaptive and Neural Computation Division for Informatics, Edinburgh University, EH 9LW, Scotland, UK, 1999, WEB: http://www.anc.ed.ac.uk/~mjo/rbf.html.

AIAA-2000-4880

NEURAL NETWORK AND RESPONSE SURFACE METHODOLOGY FOR ROCKET ENGINE COMPONENT OPTIMIZATION

Rajkumar Vaidyanathan[(1)*], Nilay Papila[(1)†], Wei Shyy[(1)‡], P. Kevin Tucker[(2)§], Lisa W. Griffin[(2)¶],

Raphael Haftka[(1)#], and Norman Fitz-Coy[(1)**]

[(1)]Department of Aerospace Engineering, Mechanics & Engineering Science,
University of Florida, Gainesville, FL 32611-6250 USA
[(2)]NASA Marshall Space Flight Centre, Huntsville, AL 35812.

ABSTRACT

The goal of this effort is to compare the performance of response surface methodology (RSM) and neural networks (NN) to aid preliminary design of two rocket engine components. A data set of 45 training points and 20 test points, obtained from a semi-empirical model based on three design variables, is used for a shear coaxial injector element. Data for supersonic turbine design is based on six design variables, 76 training data and 18 test data obtained from simplified aerodynamic analysis. Several RS and NN are first constructed using the training data. The test data are then employed to select the best RS or NN. Quadratic and cubic response surfaces have been used in RSM and radial basis neural network (RBNN) and back-propagation neural network (BPNN) in NN. Two-layered RBNN has been generated using two different training algorithms, namely, *solverbe* and *solverb*. A two-layered BPNN is generated with Tan-Sigmoid transfer function. Various issues related to the training of the neural networks have been addressed, including number of neurons, *error goals,* and *spread constants*, and the accuracy of different models in representing the design space. A search for the optimum design is carried out using a standard, gradient-based optimization algorithm over the response surfaces represented by the polynomials and trained neural networks. Usually a cubic polynomial performs better than the quadratic polynomial but exception have been noticed. Moreover, cubic polynomials requires larger amount of data for regression analysis as compared to quadratic polynomials due to more number of coefficients in the equation. Among the NN choices, the RBNN designed using *solverb* yields more consistent performance for both engine components considered. The ease of training an RBNN and the consistency in performance over BPNN does promise the possibility of it being used as an optimization strategy for engineering design problems.

* Graduate Student Assistant, Student Member AIAA
† Graduate Student Assistant, Student Member AIAA
‡ Professor and Dept. Chair., Associate Fellow AIAA
§ Aerospace Engineer, Member AIAA
¶ Aerospace Engineer, Member AIAA
# Distinguished Professor, Fellow AIAA
** Associate Professor, Member AIAA

1 INTRODUCTION

1.1 General Background

Advanced rocket propulsion systems are being proposed to meet goals for increased performance, robustness, and safety while concurrently decreasing weight and cost. These new goals are forcing consideration of design variables over ranges and in combinations not typically employed, thereby increasing the design space complexity. Objective and efficient evaluation of these new and complex designs can be facilitated by development and implementation of systematic techniques. Accordingly, Response Surface Methodology[1] (RSM) and Neural Network[2] (NN) techniques have been used to generate surrogate models representing data obtained from complex numerical and experimental simulations. An optimization algorithm is then used to interrogate these models for optimum design conditions, based on specified constraints. In this study, the preliminary design issues related to rocket propulsion components, including gas-gas injectors and supersonic turbines have been investigated. The objective of this effort is to assess relative performance of RSM and NN techniques in representing the design space.

A polynomial-based RSM is used, in which the design space is represented with quadratic and cubic polynomials in the dependent variables. The polynomial coefficients are obtained by linear regression. Then using a gradient search method the maximum or the minimum of the surface can be located. Response Surface methodologies have been used before for rocket engine component design. For example, Tucker et al.[3] have used RSM for rocket injector design. The approach is not tied to any specific data type or source. The dimensionality of the data is not a concern, and data obtained through both numerical and experimental methods can be effectively used. RSM enables the designer to combine any number of design variables for different types of injectors and propellant combinations. This generality allows the consideration of information at varying levels of breadth (i.e., scope of design variables) and depth (i.e., details of the design variables).

The RSM is effective in representing the global characteristics of the design space and it filters noise associated with design data. Depending on the order of polynomial employed and the shape of the actual response surface, the RSM can introduce substantial errors in certain regions of the design space. It has been shown by Shyy et al.[4] that for a given injector design, a third order response surface performs better than a second order surface. Generation of polynomial based surfaces can be costly for cases involving many of design variables due to the amount of data required to evaluate the coefficients. In fact, the number of coefficients increases rapidly with the order of polynomial. For example, a complete second-order polynomial of N design variables has $(N+1)(N+2)/(2!)$ coefficients. A complete cubic model has $(N+1)(N+2)(N+3)/(3!)$ coefficients. The choice of order of the polynomial and the terms to be included depends on the design problem. Many combinations of terms may have to be tried to represent the design space before the best one can be selected.

An optimization scheme requiring large amounts of data and evaluation time to generate meaningful results is of limited value. While the preliminary designs can be accomplished with empirically based information, detailed designs often require use of data from experiments and/or computational fluid dynamics (CFD) analyses. This data can be time consuming and expensive to generate in large quantities. Recently, NN have been used to represent the models instead of the more typical polynomial RSM. Work in the area of NN by Shyy et al.[4] and Papila et al.[5] have shown that some NN can perform well even when a modest amount of data is available. NN involves a linear or a nonlinear regression process, depending upon the type of neural network used, to evaluate the weights associated with the neurons of the network. Norgaad et al.[6] and Ross et al.[7] have investigated the feasibility of reducing wind tunnel test times by using NN to interpolate between measurements and demonstrated cost savings. These works have focused on using the NN to predict data. Attempts to use the network as a function evaluator and then to link it to the optimizer have been made by Protzel et al.[8], Rai and Madavan[9] and Greenman and Roth[10].

NN are highly flexible in functional form and hence can offer significant potential for representing complex functions. Networks that are flexible and employ linear regression methods can use both of these properties to improve the performance. The number of neurons in the network, size of the region over which the neuron is sensitive, and the training accuracy of the network are some of the parameters that need to be selected in a network. These can be determined by comparing the performance of NNs designed with different values of these parameters. NNs can be effectively used in two ways. First, they can be used in conjunction with RSM. In complex regions of the surface, the NN can be trained using the existing data. The trained NN can then be used to generate additional data to augment existing data, thus possibly enhancing the accuracy of the surface in that particular area. Such an approach was investigated by Shyy et al[4]. This work demonstrated that the NN could indeed yield additional information to help generate more accurate polynomial-based response surfaces. Second, NN can generate data to be used directly in conducting gradient-based optimization. In other words, NN can perform the role of either enhancing the fidelity of a polynomial-based response surface, as in the first approach, or generating information as input to an optimizer by itself without resorting to a polynomial representation, as in the second approach. Either way, the only function evaluations required are for the points sought by the optimizer, which searches the design space based on the sensitivity of the response to the perturbations in the design variables.

1.2 Scope

The present work is aimed at a direct comparison of the RSM and NN techniques in terms of accuracy and efficiency; the hybrid RSM-NN scheme noted above will not be used here. Both techniques are applied to data used in the design of two rocket engine components: a shear co-axial injector and a supersonic turbine. Variations of each technique are evaluated. Both second and third order polynomials will be used for the Response Surface (RS). Two NN schemes, radial basis and the more commonly used back-propagation NNs are used. The same database for each component will be used to train both the RS and the NN. Both will then be linked to an optimization procedure. There is little rigorous theory in the literature to establish the desired framework for a clear comparison between the performances of the two techniques. However, this work provides an assessment of the techniques regarding their practical use in the rocket engine component design process.

2 APPROACHES

2.1 Summary of Analytical Models and Design Variables

Two components of a rocket propulsion system have been considered here, the injector and the turbine. First, a shear coaxial injector element that uses gaseous oxygen (GO_2) and gaseous hydrogen (GH_2) as propellants is used to investigate the relative performance of RSM and NN in the design of rocket engine injectors. The original data set from Tucker et al[3] (45 design points) is used to generate quadratic and cubic response surfaces for both, energy release efficiency (*ERE*), a measure of injector performance, and chamber wall heat flux (Q). These 45 design points are evenly distributed over the design space. *ERE* was obtained using correlations taking into account combustor length, L_{comb} (length from injector to throat), and the propellant velocity ratio, V_f/V_o. The nominal chamber wall heat flux at a point just downstream of the injector, Q_{nom}, was calculated using a modified Bartz equation. It was then correlated with propellant mixture ratio, O/F, and propellant velocity ratio, V_f/V_o to yield the actual chamber wall heat flux, Q. The accuracy of each polynomial fit on the original data set is evaluated. Two different types of radial basis NN (RBNN) and a back propagation NN (BPNN) are also trained to represent *ERE* and Q. Each surface is then used to conduct design optimization over the same range of independent variables. The optimal design points are compared with exact points calculated from the empirical model of Calhoon et al[11]. The range of design variables considered in this study is shown in Table 1. Tables 1A, 2A and 3A (see appendix) show the performance and heat flux for the 45 combinations of O/F,

V_f/V_o and L_{comb} considered. Table 4A (see appendix) contains 20 additional data points that are not used in the generation of response surfaces or the neural networks. These points are used to assess the accuracy of different variants of RSM and NN.

The other propulsion system component examined is a supersonic turbine where the preliminary design is conducted by one-dimensional aerodynamic analysis using *FpgenML*[12]. *FpgenML* generates a flowpath and runs a preliminary meanline calculation on this flowpath. In this study, a single stage turbine has been considered. There are six design parameters and four output variables involved in this design process. There are 76 design points available for training (Table 5A). These 76 points were selected by using a face centered composite (fcc) design. Instead of 77 design points, as would be provided by a *fcc* design for six variables, only 76 were available since the meanline code could not converge for one of the designs. The design variables are the mean diameter, D, RPM, blade annulus area, A_{ann}, vane axial chord, C_v, blade axial chord, C_b, and stage reaction, k_r. These are parameters influencing the structural properties and performance of the turbine. Overall efficiency of the turbine, η, turbine weight, W, a lumped inertia measure, $(AN)^2$ ($A_{ann} \times (RPM)^2$) and speed at pitchline, V_{pitch} ($D \times RPM$) are chosen as dependent variables. The goal is to maximize the incremental payload (Δpay), which is derived from turbine weight (W) and efficiency (η). Therefore, the objective is a design where W is minimized and η is maximized. Due to the structural considerations, constraints have to be imposed on $(AN)^2$ and V_{pitch}.

Using 18 additional simulations, distributed within the design space, the accuracy of the models is tested. This additional testing data set, generated with the same analysis method, is shown in Table 6A. The ranges considered for the design variables and the dependent variables are shown in Eqs. (1) and (2).

For the design variables:

$$\left.\begin{aligned} &15.2 > D > 5.1\\ &43954.4 > RPM > 18837.6\\ &69.9 > A_{ann} > 37.6\\ &1.3 > C_v > 0.3\\ &1.3 > C_b > 0.3\\ &0 > k_r > 0.5 \end{aligned}\right\} \quad (1)$$

For dependent variables:

$$\left.\begin{aligned} &0.70 > \eta > 0.14\\ &913.62 > W > 480.58\\ &1.1644\times10^{11} > (AN)^2 > 0.182\times10^{11}\\ &2576.6 > V_{rim} > 67.4 \end{aligned}\right\} \quad (2)$$

2.2 Objective Functions

When attempting to optimize two or more different objective functions, conflicts between them arise because of the different relationships they have with the independent parameters. To solve this problem, a multi-objective[12] approach is investigated in this study. Here, competing objective functions are combined to a single composite objective function. The maximization of the composite function effectively provides a compromise between the individual functions. An average of some form is normally used to represent the composite function. For example, Tucker et al[3] used a geometric mean to combine their two objectives, ERE and Q. The composite desirability is of the form

$$D = \left(\prod_{i=1}^{l} d_i\right)^{1/l} \quad (3)$$

where D is the composite objective function, d_i's are normalized values of the objective functions and l is the number of objective functions.

Another way of constructing a composite function is to use a weighted sum of the objective functions. The composite desirability function can then be expressed as

$$D = \sum_{i=1}^{l} \alpha_i f_i \quad (4)$$

where D is the composite objective function and f_i's are the non-normalized objective functions. The α_i's are dimensional parameters that control the importance of each objective function.

For the injector, the goal is to maximize the energy release efficiency, ERE while minimizing the chamber wall heat flux, Q. This is achieved by maximizing a composite objective function given by Eq (5).

$$D = \left(d_{ERE} d_Q\right)^{1/2} \quad (5)$$

where the normalized functions are defined in Eqs. (6) and (7). In the case where a response should be maximized, such as ERE, the normalized function takes the form:

$$d_{ERE} = \left(\frac{ERE - A}{B - A}\right)^s \text{ for } A \le ERE \le B \quad (6)$$

where B is the target value and A is the lowest acceptable value. We set $d_{ERE} = 1$ for any $ERE > B$ and $d_{ERE} = 0$ for $ERE < A$. The choice of s is made based on the subjective importance of this objective in the composite desirability function. In the case where a response is to be minimized, such as Q, the normalized function takes on the form:

$$d_Q = \left(\frac{E - Q}{E - C}\right)^t \text{ for } C \le ERE \le E \quad (7)$$

where C is the target value and E is the highest acceptable value. We set $d_Q = 1$ for any $Q < C$ and $d_Q = 0$ for $Q > E$. A, B, C, and E are chosen according to the designer's priorities or, as in the present study, simply as the boundary values of the domain of ERE and Q. The value of t is again chosen to reflect the importance of the objectives in the design. In the study A and B are equal to 95.0 and 99.9, respectively. Values of C and E are equal to 0.48 and 1.1, respectively. Both s and t were set to a value of 1.

In the case of the turbine, a weighted sum of the two objectives η and W has been used. The expression, in the context of the turbine gives the incremental value of the payload with the change in W and η. The goal is to maximize this incremental value, which in turn results in minimum W and maximum η.

$$D = \Delta pay = 80 \times 100 \times (\eta - \eta_b) - (W - W_b) \tag{8}$$

where η = the calculated efficiency
η_b = the baseline efficiency (η_b=0.627)
W = calculated weight
W_b = the baseline weight (W_b=1140.0lbs).

The weight associated with η expressed in percentage, by multiplying it with 100, is 80 and the weight associated with W is –1. This relationship is developed based on detailed turbopump design processes. For one percent increase in efficiency a payload increase of 80lbs can be achieved, and as the weight of the turbine increases the payload has to be correspondingly decreased.

2.3 Response Surface Methodology (RSM)

Polynomial RSM constructs polynomials of assumed order and unknown coefficients based on regression analysis. The solution for the set of coefficients that best fits the training data is a linear least square problem. The number of coefficients to be evaluated depends on the order of polynomial and the number of design parameters involved.

According to the injector model developed by Calhoon et al[11], injector performance, as measured by *ERE* depends only on the velocity ratio, V_f/V_o, and combustion chamber length, L_{comb}. Examination of the original data sets in Tables 1A-3A (see appendix) indicates 15 distinct design points for *ERE*. Since chamber wall heat flux depends only on the velocity ratio, V_f/V_o, and the oxidizer to fuel ratio, *O/F*, there are 9 distinct design points for Q. The design space for this problem is depicted in Figure 1. For *ERE*, the 5 distinct chamber lengths offer the potential for a fourth-order polynomial fit in L_{comb}, while the three different velocity ratios limit the fit in V_f/V_o to second order. Quadratic and cubic response surfaces for both *ERE* and Q have been generated for evaluation. The above-noted limitations on the data limits the cubic surfaces to be third order in L_{comb} only.

As already mentioned, to construct a complete quadratic polynomial of N design variables, the number of coefficients required is $(N+1)(N+2)/(2!)$. In the turbine case with 6 design variables, we would need to estimate 28 coefficients. A complete cubic model would require $(N+1)(N+2)(N+3)/(3!)$ or 84 coefficients and four levels. Since the data available is not sufficient to evaluate all the cubic terms, reduced cubic models are employed.

The response surfaces were generated by standard least-squares regression using JMP[14], a statistical analysis software package. JMP is an interactive, spreadsheet-based program having a variety of statistical analysis tools. Statistical techniques are also available for identifying polynomial coefficients that are not well characterized by the data. A stepwise regression procedure based on t-statistics is used to discard terms and improve the prediction accuracy. The t-statistic, or t-ratio, of a particular coefficient is given by the value of the coefficient divided by the standard error of the coefficient, which is an estimate of its standard deviation. The accuracy of different surfaces at points different from the training data can be estimated by comparing the adjusted root mean square error defined as:

$$\sigma_a = \sqrt{\frac{\sum e_i^2}{n - n_p}} \tag{9}$$

Here e_i is the error at i^{th} point of the training data, n is the number of training data points and n_p is the number of coefficients. When the data contains uncorrelated Gaussian noise, σ_a provides an unbiased estimate of that noise. Even when the error is not solely due to noise σ_a provides a good overall comparison among the different surface fits.

The accuracy of the models in representing the objective functions is also gauged by comparing the values of the objective function at test design points, different from those used to generate the fit. The root mean square error, σ, for the test set is given by:

$$\sigma = \sqrt{\frac{\varepsilon_i^2}{m}} \tag{10}$$

In this equation ε_i is the error at the i^{th} test point and m is the number of test points.

2.4 Neural Networks

Two different types of NN have been used, namely radial basis[15] and back-propagation[15]. The training process of the network is a cyclic process and the weights and biases of the nodes of the network are adjusted until an accurate mapping is obtained. This trained network can then predict the values of the objective for any new set of design variables in the design space. The neural network toolbox[15] available in *Matlab* is used for the current analysis.

2.4.1 Radial Basis Neural Networks (RBNN)

Radial-basis neural networks are two-layer networks with a hidden layer of radial-basis transfer function and a linear output layer (Figure 2). RBNN requires large number of neurons, depending on the size of the data set, but they can be designed in a small amount of time. This is due to the fact that the process of determining the weights associated with the large number of neurons uses linear regression. Thus, they may be efficient to train when there are large amounts of data available for training.

In *Matlab*, radial-basis networks can be designed using two different design procedures, *solverbe* and *solverb*. *Solverbe* designs a network with zero error on the training vectors by creating as many radial basis neurons as there are input sets. Therefore, *solverbe* may result in a larger network than required and map the network exactly, thereby fitting numerical noise. A more compact design in terms of network size is obtained from *solverb,* which creates one neuron at a time to minimize the number of neurons required. At each epoch or cycle, neurons are added to the network till a user specified RMS error is reached or until the network has the maximum number of neurons possible. The design parameters for *solverb* are the *spread constant*, a user defined RMS *error goal*, and the maximum number of epochs whereas it is only the *spread constant* for *solverbe*.

The transfer function for radial basis neuron is *radbas*, which is shown in Figure 2b. *Radbas* has maximum and minimum outputs of 1 and 0, respectively. The output of the function is given by

$$a = radbas\left(dist\left(w, p\right) \times b\right) \tag{11}$$

where *radbas* is the transfer function, *dist* is the vector distance between the network weight vector, $\boldsymbol{w}$ and the input vector, $\boldsymbol{p}$, and b is the bias. The weights associated with each neuron in the network together comprise the weight vector. The input vector is the set of design points that are used to train the network. In a radial basis network (Figure 2a) each neuron in the *radbas* hidden layer is assigned weights, w_1 which are equal to the values of different input design points. Therefore, each neuron acts as a detector for a different input. The bias for each neuron in that layer, b_1 is set to 0.8326/*sc*, where *sc* is the spread constant, a value defined by the user. This defines the area of response of each neuron. The whole process is then reduced to the evaluation of the weights, w_2,and biases, b_2, in the output linear layer, which is a linear regression problem. If the input to a neuron is identical to the weight vector, the output of that neuron is 1, since the effective input to the transfer function is zero. When a value of 0.8326 is passed through the transfer function the output is 0.5. For a vector distance equal to or less than 0.8326/*b*, the output is 0.5 or more. The *spread constant* controls the efficiency of the RBNN. It defines the radius of the design space over which a neuron has a response of 0.5 or more. Small values of *sc* can result in poor response in a domain not closely located to neuron positions, that is, for inputs that are much different from the weights assigned to a neuron as compared to the defined radius the response from the neuron will be negligible. This would prevent the network from assessing the trend of the design space accurately causing an underfitting of the domain. Large values will result in low sensitivity of neurons. Since the radius of sensitivity is large, neurons whose weights are different from the input values by a large amount will also have high output thereby resulting in an equivocal response from all the neurons and this might result in overfitting, that is, the noise in data may also be fit accurately. Hence, the network may not be able to account for the characteristics of the data. Both conditions would result in a badly designed network. The best value of the *spread constant* for some test data can be found by comparing σ for networks with different *spread constants*. In case of the injector design there are two objectives, namely *ERE* and *Q* and for turbine the objectives are η and *W*. Figures 3 and 4 give the variation of σ for the network design with *solverbe* for the objective functions of the two engine components. In case of *solverb* the *error goal* during training also defines the accuracy of the network. An objective of fitting a numerical model is to remove the noise associated with the data. A model, which maps exactly as *solverbe* does, will not eliminate the noise, whereas *solverb* will. Figures 5 and 6 give the variation of σ for the network design with *solverb* for the objective functions of the two engine components.

Comparing Figures 3-6 it can be seen that for low values of *spread constant* the NN network has a poor performance. As the *spread constant* increases σ asymptotically decreases. However, as demonstrated by Figure 5a the performance of the network can deteriorate for higher values of the *spread constant.* The region with a large variation in σ is highly unreliable because this indicates a high sensitivity of the model to a small variation of *spread constant* and possibly the test data, in this region. Hence the desirable *spread constant* is selected from the region where the performance of the network is relatively consistent.

Figures 5 and 6 show also the influence of *error goal* on the network. Generally if a network maps the training data accurately it can be expected to perform efficiently with the test data. However, accurately mapping noisy data may result in poor prediction capabilities for the network. The variation in the performance is not significant except for the *ERE* and *Q* network (Figure 5), where the poor performance of the network at high values of *spread constant* improves for a larger *error goal.* This may indicate the presence of noise in the data for *ERE*, which *solverb* is able to eliminate with an appropriate *error goal.* Figure 7 shows variations in number of epochs and σ with the variation of *error goal* for a given *spread constant* when RBNN is designed with *solverb.* The number of neurons in the network is one more than the number of epochs. One expects that as the *error goal* increases the number of epochs becomes smaller and the network performs less accurately as in Figures 7a and 7b. However as demonstrated by Figures 7c and 7d, a more stringent *error goal* for the training data does not necessarily result in better predictive capability against the test data. Less accurate network can be designed for these objectives, which have smaller prediction error.

When choosing an appropriate network the above-mentioned features have to be considered. The performance of the constructed NN is best judged by comparing the prediction error as given in Eq. (10), for different networks. Using *solverbe*, networks are designed with varying *spread constants* and the one that yields the smallest error is selected. When *solverb* is used, networks are designed for different *spread constants* and *error goals*. The network that gives the smallest error for the test data is used. The details of the networks selected are discussed in later sections.

2.4.2 Back-propagation Neural Networks (BPNN)

Back-propagation networks are multi-layer networks with hidden layers of sigmoid transfer function and a linear output layer (Figure 8). The transfer function in the hidden layers should be differentiable and thus, either log-sigmoid or tan-sigmoid functions are typically used. In this study, a single hidden layer with a tan-sigmoid transfer function, *tansig*, (Figure 8b) is considered. The output of the function is given by

$$a = tansig\left(w \cdot p + b\right) \tag{12}$$

where *tansig* is the transfer function, w is the weight vector, p is the input vector and b is the bias vector. The maximum and minimum outputs of the function are 1 and –1, respectively.

The number of neurons in the hidden layer of a back-propagation network is a design parameter. It should be large enough to allow the network to map the functional relationship, but not too large to cause overfitting. Once it has been chosen, the network design is reduced to adjusting the weight matrices and the bias vectors. These parameters for back-propagation networks are usually adjusted using gradient methods like the *Levenberg-Marquardt*[15] technique. In *Matlab*, back-propagation networks can be trained by using three different training functions, *trainbp, trainbpx* and *trainlm*. The first two are based on the steepest descent method. Simple back-propagation with *trainbp* is usually slow since it requires small learning rates for stable learning. *Trainbpx,* applying momentum or adaptive learning rate, can be considerably faster than *trainbp*, but *trainlm,* applying *Levenberg-Marquardt* optimization, is the most efficient since it is based on a more efficient optimization algorithm.

The design parameters for *trainlm* are the number of neurons in the hidden layer, a user defined *error goal*, and the maximum number of epochs. The training continues until either the *error goal* is reached, the minimum error gradient occurs or the maximum number of epochs has been met.

For BPNN, the initial weights and biases are randomly generated and then the optimum weights and biases are evaluated through an iterative process. The weights and biases are updated by changing them in the direction of down slope with respect to the sum-squared error of the network, which is to be minimized. The sum-squared error is the sum of the squared error between the network prediction and the actual values of the output. In BPNN (Figure 8a) the weights, w_1, and biases, b_1, in the hidden *tansig* layer are not fixed as in the case of RBNN. Hence, the weights have a nonlinear relationship in the expression between the inputs and the outputs. This results in a nonlinear regression problem, which takes a longer time to solve than RBNN. Depending upon the initial weights and biases, the convergence to an optimal network design may or may not be achieved. Due to the randomness of the initial guesses, if one desires to mimic the process exactly for some purpose, it is impossible to re-train the network with the same accuracy or convergence unless the process is reinitiated exactly as before. The initial guess of the weights is a random process in Matlab. Hence to re-train the network the initial guess has to be recorded.

The architecture is decided based on past experience with similar kind of dataset. For a given objective the *error goal* is fixed and the number of hidden layer neurons are varied between 2 and the total number of inputs. Each network is retrained few times so as to start the search from random initial weights and biases. The networks that do not achieve the *error goal* are discarded. Among the converged networks the selection of the best network is made based on the value of σ. The goal is to attain as low a value for σ as possible. The number of neurons in the hidden layer is increased one at a time till the error goal is achieved and a small value of σ is obtained. Although this method may not be the best way to obtain the best BPNN, it is considered adequate for the current study. At times larger network has a high value of σ, which maybe due to overfitting of the design space. To prevent the model from converging to a local minimum, an iterative method is used as suggested by Stepniewski et al[16]. The obtained network is retrained with initial weights obtained by perturbing the weights of the obtained network.

$$w = w_o + \lambda r w_o \tag{13}$$

where $\boldsymbol{w}$ is the initial weight vector for the network to be trained, $\boldsymbol{w_o}$ is the weight vector of the obtained network, λ is the level of pertubation (0.1) and $\boldsymbol{r}$ is a matrix of random numbers between –1 to 1.

2.5 Optimization Process

The entire optimization process can be divided into two parts:

1) RS/NN training phase for establishing an approximation,
2) Optimizer phase.

In the first phase, RS or NN are generated with the available training data set. In the second phase the optimizer uses the RS/NN during the search for the optimum until the final converged solution is obtained. The initial set of design variables is randomly selected from within the design space. The flowchart of the process is shown in Figure 9.

The optimization problem at hand can be formulated as $min\{f(x)\}$subject to $lb \le x \le ub$, where lb is the lower boundary vector and ub is the upper boundary vector of the design variables vector x. If the goal is to maximize the objective function then $f(x)$ can be written as $-g(x)$, where $g(x)$ is the objective function. Additional linear or nonlinear constraints can be incorporated if required. The present design process does not have any such additional constraints. The optimization toolbox[17] in *Matlab* used here employs a sequential quadratic-programming algorithm.

3 RESULTS AND DISCUSSION

The RS and NN are constructed using the training data. The test data is then employed to select the best RS or NN. Specifically in RSM, the difference between the RS and the training data, as given by Eq. (9), is normally used to judge the performance of the fit. The additional use of the test data helps to evaluate the performance of different polynomials over design points not used during the training phase. This gives a complementary insight into the quality of the RS over the design space. For both the rocket engine components, different polynomials were tried. Table 2 compares the performance of different polynomials used to represent the two objective functions of the injector case, *ERE* and *Q*. Starting with the all the possible cubic terms in the model, revised models are generated by removing and adding terms. Similar kind of analysis is also done for the turbine case. The best polynomial is selected based on a combined evaluation between σ_a and σ.

For the NN, the test data helps evaluate the accuracy of networks with varying neurons in BPNN and varying spread constant in RBNN. Thus the test data are part of the evaluation process to help select the final NN. Based on the RSM or NN model, a search for optimum design is carried out using a standard, gradient-based optimization algorithm over the response surfaces represented by the polynomials and trained neural networks.

3.1 Shear-Coaxial Injector

According to the available data, the injector performance, *ERE*, depends only on the velocity ratio, V_f/V_o, and combustion chamber length, L_{comb}, which indicates 15 distinct design points for *ERE*. The chamber wall heat flux, *Q*, depends on velocity ratio, V_f/V_o, and oxidizer to fuel ratio, *O/F*, and has nine distinct points. For *ERE*, as seen from Figure 1, five distinct levels for L_{comb} offers the potential for a fourth-order polynomial fit in the same, while three different velocity ratios and oxidizer to fuel ratio limit the fit in these variables to second order.

A reduced quadratic and a higher order response surfaces are used for the two objective functions. The first model in Table 2a and the sixth model in Table 2b are the selected cubic models for *ERE* and *Q*, respectively. There is no considerable improvement notice among the remaining cubic model for *ERE*. For *Q*, the selected model is the best in terms of σ_a, although there are other models with identical value of σ.

$$ERE = 70.43 + 1.580V_f/V_o + 6.208L_{comb} - 0.190(V_f/V_o)L_{comb} - 0.331(L_{comb})^2 \tag{14}$$

$$Q = 0.479 - 0.046O/F + 0.191V_f/V_o + 0.009(O/F)^2 - 0.028(O/F)V_f/V_0 \tag{15}$$

$$\begin{aligned}ERE &= 50.059 + 3.758V_f/V_o + 14.573L_{comb} \\ &-0.05(V_f/V_o)^2 - 0.777(V_f/V_o)L_{comb} \\ &-1.459(L_{comb})^2 + 0.002(V_f/V_o)^2 L_{comb} \\ &+0.046V_f/V_o(L_{comb})^2 + 0.047(L_{comb})^3\end{aligned} \tag{16}$$

$$\begin{aligned}Q &= -0.566 - 0.358O/F + 0.383V_f/V_o \\ &-0.0191(O/F)^2 - 0.107(O/F)V_f/V_o \\ &-0.003(V_f/V_o)^2 + 0.005(O/F)^2 V_f/V_o \\ &+0.002(O/F)(V_f/V_o)^2\end{aligned} \tag{17}$$

Equations (14) and (15) are the reduced quadratic responses and Eqs. (16) and (17) represent the reduced cubic polynomials used for the two objective functions. The t-statistics for the coefficients in Eq. (14) vary between 49.30 and 8.06. For the coefficients in Eq. (15), they vary between 6.28 and 0.52. In Eqs. (16) and (17), the t-statistics of the coefficents vary between 14.69 and 0.31 and 3.36 and 0.74, respectively.

The radial basis networks designed with solverbe are the largest with 15 neurons in the hidden layer for *ERE* network and nine neurons for the *Q* network. *Solverb* designs a network for *ERE* with 14 neurons in the hidden layer and a network for *Q* with eight neurons. Compared to RBNN, BPNN has fewer neurons, the number of neurons in the hidden layer are eight and four for the *ERE* and *Q* networks, respectively. Details of the networks used are listed in Table 3. The *spread constant* used for RBNNs and the *error goal* of the training data is also given in Table 3. The *spread constant* values are selected from the region where the performance of the network is consistent with the variation of *spread constant* (Figures 3-6). The *error goal*, in the case of *solverb*, is selected based on the network with the best performance for the ideal *spread constant* (Figure 7).

The error in predicting the values of the objective function by different schemes is given in Table 4. Several observations can be readily made.

1. The NN method performs better than the RSM for this data set.
2. Both *solverbe* and *solverb* are of comparable performance.
3. The BPNN helps generate a smaller network it and hence performance at par as compared to RBNN.
4. The polynomial-based response surfaces are not as flexible as the NN. However, the cubic polynomial is more accurate than the quadratic one.

The various models generated are compared with test data in Figures 10 and 11. The curves representing the NN predictions are closer to the data obtained from the injector model than the RSs thereby demonstrating that NN models are able to predict better than the RSs. BPNN performs as well as RBNN but tends to be flat. Due to its lower order, the quadratic polynomial is flat. The cubic polynomial is able to perform better than quadratic.

The optimum solution obtained from various schemes is shown in Table 5 and Figures 12 and 13. The aim is to maximize *ERE* and minimize *Q*. The trend of the objective functions in the design space is monotonic and hence every model is able to select identical optimum design for the given constraints. The flatness of the polynomials result in bad predictive values of the objective function for the optimum design. The cubic polynomial is more flexible than quadratic but is not consistent. For a V_f/V_o constraint of 4 the quadratic polynomial is more accurate but for higher values of V_f/V_o the cubic polynomial is more accurate. In contrast, the NN models are able to perform well. Since the optimum design happens to be the same as one of the training points, *solverbe* is able to predict the values of the objective function accurately. *Solverb* performs equally well, thereby showing the capability of performance with fewer neurons. Performance of BPNN is not as satisfactory as suggested in Table. 4. For lower constraints of V_f/V_o, it performs poorly but for higher values of V_f/V_o it is good. This may be due to the selection of fewer neurons in the hidden layers of the networks. Overall, it is still better than to the RSM and demonstrates the flexibility of NN over RS.

As stated by Papila et al[5], when it comes to choosing between NN and polynomials, polynomials are easy to compute. The number of coefficients might be numerous but the linearity of the system expedites the process of coefficient evaluations. This is also the reason RBNN train fast. On the other hand, the weights of BPNN are evaluated through a nonlinear process. Of all the NN presented here, the one designed with the help of *solverbe* is the fastest to train since the values of the weights are set to values of the input dependent variables. *Solverb* trains with the addition of one neuron at a time with weights similar to the input and hence is slower. BPNN is slower to train than RBNN because at each step the error is propagated back to all the weights in the system unlike the RBNN.

3.2 Supersonic Turbine

The generation of RS and the training of the NNs are done with the 76 design points in Table 5A. The analysis was initially done without the constraints and then with the constraints on $(AN)^2$ and V_{pitch}.

A quadratic RS was initially generated. Then, cubic terms were included. Cubic terms that are products of three different variables were included because of the number of data available and the number of levels being three. The trend of the design data also suggests the presence of some of these terms. Therefore, the initial cubic equation has 45 terms. A reduced third order RSs for η and W was selected based on the relative performances of different polynomials obtained by removing terms from the initial cubic equation based on t-statistics. The cubic equation was selected based on the evaluated value of σ_a and σ. Table 6 suggests that the reduced cubic polynomial is better than the quadratic polynomial since σ_a is better for the former. The values of σ are comparable.

$$\begin{aligned}
\eta &= 0.410+0.180D+0.112RPM+0.012A_{ann} \\
&+0.006C_v+0.003C_b-0.025k_r-0.026D^2 \\
&+0.032D\cdot RPM-0.010RPM^2+0.006DA_{ann} \\
&+0.005A_{ann}RPM+0.002C_vD+0.003C_vRPM \\
&-0.007C_v^2-0.006C_b^2-0.010k_rD-0.006k_rRPM \\
&-0.005k_rA_{ann}-0.005k_rC_v-0.002k_rC_b-0.008k_r^2 \\
&+0.002DA_{ann}RPM+0.001DC_vRPM-0.002DA_{ann}k_r \\
&-0.003DC_vk_r-0.002A_{ann}RPMk_r \\
&-0.002C_bRPMk_r \qquad (18)
\end{aligned}$$

$$\begin{aligned}
W &= 734.123+169.310D-34.926RPM \\
&+13.313A_{ann}+6.756C_v+4.134C_b-29.062k_r \\
&-42.01D^2-25.157D\cdot RPM+4.690DA_{ann} \\
&-5.131A_{ann}^2+1.261C_vD+0.512C_vRPM \\
&-6.706C_v^2-1.023C_bD-0.746C_bRPM \\
&+0.236C_bA_{ann}-5.366C_b^2-7.695k_rD \\
&+3.396k_rRPM-4.803k_rA_{ann}-4.851k_rC_v \\
&-1.833k_rC_b-7.676k_r^2+0.437DC_vRPM \\
&-0.511DC_bRPM+3.292Dk_rRPM+0.627DA_{ann}C_b \\
&-1.443DA_{ann}k_r-2.392DC_vk_r \\
&-0.539C_vRPMk_r \qquad (19)
\end{aligned}$$

The t-statistics for the coefficients in Eq. (18) varies between 179.72 and 1.2. The coefficients in Eq. (19) have t-statistics varying between 822.66 and 0.68.

The networks designed with *solverb* have 37 and 75 neurons for η and W, respectively in the hidden layer, while those designed with *solverbe* has 76 neurons each. The BPNN uses significantly less number of neurons by generating networks with five and 60 neurons for η and W, respectively, in a single hidden layer. The NN architectures chosen are listed in Table 7.

The accuracy of the various models is tested with the data available in Table 6A and the error is shown in Table 8. *Solverbe* has a poor prediction for η, which might be due to overfitting, but performs well for W. The outcome of Table 8 for the supersonic turbine is similar to that of Table 4 for injector, except that BPNN is clearly inferior to RBNN. Overall, based on the two cases, it seems that *solverb* is most consistent among all methods evaluated.

The optimum solutions subjected to the constraints, of $(AN)^2$ limited to less than $6.0 \times 10^{10} in^2 rpm^2$ and V_{pitch} is limited to less than 1600.0*in/sec*, are presented in Table 9. Since $(AN)^2$ is proportional to the product of square of *RPM* and A_{ann}, and V_{pitch} is proportional to D times *RPM*, no NN/RS is generated for them. By comparing the predicted optimal design by the various methods, one observes that *solverbe* and BPNN yield noticeably larger errors in η and W, respectively. *Solverb* and the response surface are more consistent with both η and W. Judged by the error in predicting Δpay, it seems that the RSM is most accurate. However, since the real goal is to maximize Δpay, it is important to note that the actual value of Δpay for the optimal design chosen by the RSM is the worst. Clearly, the large multiplier in Eq. (8) causes bias in relative weighting between η and W, which in turn, causes different "apparent" accuracy levels by various methods.

From a design perspective, it is interesting to understand the impact of the constraints from A_{ann} and V_{pitch} on the optimal turbine parameters. Such an assessment is offered in Figures 14 and 15. As *D*, *RPM* and A_{ann} decrease, η, W, V_{pitch}, AN^2 and Δpay decrease. C_b and C_v are almost constant over the design space and they do not have any noticeable effect on the objective functions and constraints. In the case of *Cv*, the BPNN shows a small perturbation for the analysis with the constraint. This might be due to the mapping of some noise by BPNN. Otherwise it is unaffected by the inclusion of the constraints. The stage reaction, K_r, is unaffected as expected, since we are dealing only with the single stage of the turbine. Hence there is no split on the stage reaction.

4 SUMMARY AND CONCLUSIONS

In the present study, the RS and NN are first constructed using the training data. The test data are then employed to assess the performance of various polynomials and to offer insight into model improvement by removing and adding terms. The best polynomial is selected based on a combined evaluation between σ_a and σ. For the NN, the test data helps evaluate the accuracy of networks with varying neurons in BPNN and varying spread constants in RBNN. Thus the test data are adopted to help select appropriate RSM and NN models. Once an RSM or NN model is constructed, a search for optimum design is carried out using a standard, gradient-based optimization algorithm over the response surfaces represented by the polynomials and trained neural networks.

Based on the results obtained, we have reached the following conclusions.

1. Higher order polynomials perform better than lower order polynomials as they have more flexibility. However, appropriate statistical measure needs to be taken to determine the best order to use.
2. In the present study, both NN and RSM can perform comparably for modest data sizes.
3. Among all the NN configurations, RBNN designed with solverb seems to be more consistent in performance for both injector and turbine cases.
4. Radial basis networks, even when designed efficiently with *solverb*, tend to have many more neurons than a comparable back-propagation with tan-sigmoid or log-sigmoid neurons in the hidden layer. The basic reason for this is the fact that the sigmoid neurons can have outputs over a large region of the input space, while radial basis neurons only respond to relatively small regions of the input space. Thus, larger input spaces require more radial basis neurons for training.
5. Configuring a radial basis network often takes less time than that for a back-propagation network because the training process for the former is a linear in nature.
6. RBNN with the combined feature of flexibility and linear regression is more accurate than BPNN, which is nonlinear.

Based on the results shown in Tables 4 and 8, it is seen that the RBNN technique performs consistently, and holds promise for the design/optimization of advanced rocket propulsion components. The method adopted here to generate BPNN is not necessarily the most efficient. Given a better method of making the selection of the number of neurons in the hidden layer, BPNN, might be able to perform better. Future work would be aimed at implementing a better designing procedure for back-propagation networks. The work has been carried out with modest data sizes and the training is fast for such cases. Issues related to the number of design variables and training data size are critical for practical design applications, and should be addressed in the future.

5 ACKNOWLEDGMENT

The present study has been supported by NASA Marshall Space Flight Center.

6 REFERENCE:

1. Myers, R. H. and Montgomery, D. C., *Response Surface Methodology – Process and Product Optimization Using Designed Experiment*, John Wiley & Sons, 1995.
2. Hertz, J., Krogh, A. and Palmer, R. G., *Introduction to the Theory of Neural Computation, Lecture Notes Volume 1*, Addison-Wesley Publishing Company, 1992.
3. Tucker, P. K., Shyy, W. and Sloan, J. G., "*An Integrated Design/Optimization Methodology For Rocket Engine Injectors*," 34th AIAA / ASME / SAE / ASEE Joint Propulsion conference and Exhibit, July 13-15, 1998 AIAA-98-3513, Cleveland, OH.
4. Shyy, W., Tucker, P. K. and Vaidyanathan, R., "*Response Surface and Neural Network techniques for Rocket Engine Injector Optimization*," 35th AIAA / ASME / SAE / ASEE Joint Propulsion Conference and Exhibit, June 20-24, 1999, AIAA-99-2455, Los Angeles, CA.
5. Papila, N., Shyy, W., Fitz-Coy, N. and Haftka, R. T., "*Assessment of Neural Net and Polynomial-Based Techniques for Aerodynamic Applications*," 17th Applied Aerodynamics Conference, June 28-July 1, 1999, AIAA-99-3167, Norfolk, VA.
6. Norgaard, M., Jorgenson, C. C. and Ross, J. C., "*Neural Network Prediction of New Aircraft Design Coefficients*," NASA TM-112197, 1997.
7. Ross, J. C., Jorgenson, C. C. and Norgaard, M., "*Reducing Wind Tunnel Data Requirements Using Neural Networks*," NASA TM-112193, 1997.
8. Protzel, P. W., Palumbo, D. L. and Arras, M. K., "*Fault Tolerance of Artificial Neural Networks with Applications in Critical Systems*," NASA Technical Paper 3187.
9. Rai, M. M and Madavan, N. K., "*Aerodynamic Design Using Neural Networks*," AIAA Paper No. 98-4928.
10. Greenman, R. M. and Roth, K. R., "*High-Lift Optimization Design Using Neural Networks on a Multi-Element Airfoil*," Proceedings of DETC 98, ASME 1998 Computers in Engineering Conference.
11. Calhoon, D., Ito, J. and Kors, D., "*Investigation of Gaseous propellant Combustion and Associated Injector-Chamber Design Guidelines*," Aerojet liquid rocket company, NASA Cr-121234, Contract NAS3-13379, July 1973.
12. Papila, N., Shyy, W., Griffin, L. W., Huber, F. and Tran, K., "*Preliminary Design Optimization For A Supersonic Turbine For Rocket Propulsion*," 36th

AIAA/ASME/SAE/ASEE Joint Proplusion Conference and Exhibit, Paper No. 2000-3242 (2000), Huntsville, AL.
13. Haftka, R. T., Gurdal, Z. and Kamat, M. P., *Elements of Structural Optimization*, 3rd edition, Kluwer Academic Publishers, 1990.
14. SAS Institute Inc. (1995). *JMP version 3.* Cary, NC.
15. Dernuth, H. and Beale M., *Matlab Neural Network Toolbox*, The Math Works Inc, 1992.
16. Stepniewski, S. W., Greenman, R. M., Jorgensen, C. C. and Roth, K. R., "*Designing Compact Feedforward Neural Models with Small Training Data Sets*," 38th AIAA Aerospace Sciences Meeting and Exhibit, January 10-13, 2000.
17. Coleman, T., Branch, M. A. and Grace, A., *Optimization Toolbox for Use with Matlab*, Version 2, The Math Works Inc, 1999.

O/F	V_f/V_o	L_{comb}, in.
4,6,8	4	4,5,6,7,8
4,6,8	6	4,5,6,7,8
4,6,8	8	4,5,6,7,8

Table 1: Range of design variables considered for the shear coaxial injector element.

Model #	Coefficient = 0	Terms Removed	Terms Included	σ_a (%)	σ (%)
1		$(V_f/Vo)^2$	Quadratic and less	0.218	0.280
2	V_f/Vo^3			0.0857	0.212
3	V_f/Vo^3	$(V_f/Vo)^2L_{comb}$		0.0799	0.214
4		$(V_f/Vo)^2L_{comb}$, $(V_f/Vo)^3$		0.0799	0.214
5		$(V_f/Vo)^2L_{comb}$, $(V_f/Vo)^3$	$(L_{comb})^4$	0.0859	0.213
6		$(V_f/Vo)^2L_{comb}$, $(V_f/Vo)^3$	$(L_{comb})^4$, $(V_f/V_o)^2(L_{comb})^2$	0.0936	0.212
7		$(V_f/Vo)^2L_{comb}$, $(V_f/Vo)^3$	$(L_{comb})^4$, $(V_f/V_o)^2(L_{comb})^2$, $V_f/Vo(L_{comb})^3$	0.0988	0.212

Table 2(a): Different cubic polynomials for *ERE*. (Dependent variables: V_f/V_o and L_{comb}, 15 training points, 10 test points) (the error are given in percentages of the mean value of the responses).

Model #	Coefficient = 0	Terms Removed	Terms Included	σ_a (%)	σ (%)
1		$(O/F)^2$	Quadratic and less	5.445	3.490
2	$(V_f/Vo)^3$,$(O/F)^3$			5.584	2.234
3	$(O/F)^3$	$(V_f/Vo)^2$		5.584	2.094
4		$(V_f/Vo)^2$, $(O/F)^3$		5.584	2.094
5		$(V_f/Vo)^3$, $(O/F)^3$		5.584	2.234
6		$(V_f/Vo)^3$, $(O/F)^3$, $(V_f/Vo)^2$		3.909	2.094
7		$(V_f/Vo)^3$, $(O/F)^3$, $(V_f/Vo)^2$	$(V_f/Vo)^2(O/F)^2$	5.584	2.094

Table 2(b): Different cubic polynomials for Q. (Dependent variables: O/F and V_f/V_o, 9 training points, 4 test points) (the error are given in percentages of the mean value of the responses).

Scheme	# of Layers	# of neurons in the hidden layer		# of neurons in the output layer		*Error goal* aimed for during training	
		ERE	*Q*	*ERE*	*Q*	*ERE*	*Q*
RBNN (*Solverbe*)	2	15	9	1	1	0.0 {*sc* = 3.25)	0.0 {*sc* = 1.20}
RBNN (*Solverb*)	2	14	8	1	1	0.001 {*sc* = 1.05}	0.001 {*sc* = 1.05}
BPNN	2	8	4	1	1	0.01	0.01

Table 3: Neural Network architectures used to design the model for shear coaxial injector element. {*sc = spread constant*}

Scheme	σ for *ERE* (%)	σ for *Q* (%)
RBNN (*Solverbe*)	0.207	1.396
RBNN (*Solverb*)	0.133	1.536
BPNN	0.180	0.832
Partial Cubic RS	0.213	2.234
Quadratic RS	0.280	3.490

Table 4: RMS error in predicting the values of the objective function by various schemes for the shear coaxial injector element (the error are given in percentages of the mean value of the responses).

V_f/V_o	Scheme	O/F	L_{comb}, in.	*ERE*, %	Q, Btu/in^2-sec
4	RBNN (*Solverbe*)	8.0	7.0	98.60 (0.00)	0.588 (0.00)
	RBNN (*Solverb*)	8.0	7.0	98.60 (0.00)	0.588 (0.00)

	BPNN	8.0	6.9	98.64 (0.14)	0.578 (1.70)
	Partial Cubic RS	8.0	7.0	98.61 (0.01)	0.594 (1.02)
	Quadratic RS	8.0	7.0	98.67 (0.07)	0.591 (0.51)
	Model	8.0	7.0	**98.60**	**0.588**
	Model	8.0	6.9	**98.50**	**0.588**
6	RBNN (*Solverbe*)	8.0	7.0	99.20 (0.00)	0.512 (0.00)
	RBNN (*Solverb*)	8.0	7.0	99.20 (0.00)	0.512 (0.00)
	BPNN	8.0	7.0	99.18 (0.02)	0.513 (0.20)
	Partial Cubic RS	8.0	7.0	99.15 (0.05)	0.500 (2.34)
	Quadratic RS	8.0	7.0	99.17 (0.03)	0.531 (3.71)
	Model	8.0	7.0	**99.20**	**0.512**
8	RBNN (*Solverbe*)	8.0	7.0	99.40 (0.00)	0.493 (0.00)
	RBNN (*Solverb*)	8.0	7.0	99.40 (0.00)	0.493 (0.00)
	BPNN	8.0	7.0	99.41 (0.01)	0.500 (1.42)
	Partial Cubic RS	8.0	7.0	99.42 (0.02)	0.499 (1.22)
	Quadratic RS	8.0	7.0	99.67 (0.27)	0.471 (4.46)
	Model	8.0	7.0	**99.40**	**0.493**

Table 5: Optimal Solutions for fixed values of V_f/V_o and given range of *O/F* and L_{comb} obtained with NN and RSM schemes for the shear coaxial injector element. (Constraints: $4 \le O/F \le 8$, $4 \le L_{comb} \le 7$) (error given in pararenthesis for each prediction is in %)

Type of RS	σ_a for η (%)	σ for η (%)	σ_a for W (%)	σ for W (%)
Quadratic RS	2.507	0.863	0.788	1.281
Reduced Cubic RS	1.949	1.031	0.402	1.223

Table 6: Training and predicting error for different response surfaces of the objective functions of the supersonic turbine. (the error are given in percentages of the mean value of the responses)

Scheme	# of Layers	# of neurons in the hidden layer		# of neurons in the output layer		*Error goal* aimed for during training	
		η	W	η	W	η	W
RBNN (*Solverbe*)	2	76	76	1	1	0.0 {$sc = 9.50$}	0.0 {$sc = 9.45$}
RBNN (*Solverb*)	2	37	75	1	1	0.001 {$sc = 6.50$}	0.001 {$sc = 8.35$}
BPNN	2	5	60	1	1	0.001	0.001

Table 7: Neural Network architectures used to design the models for η, W and V_{rim} of the supersonic turbine. {*sc = spread constant*}.

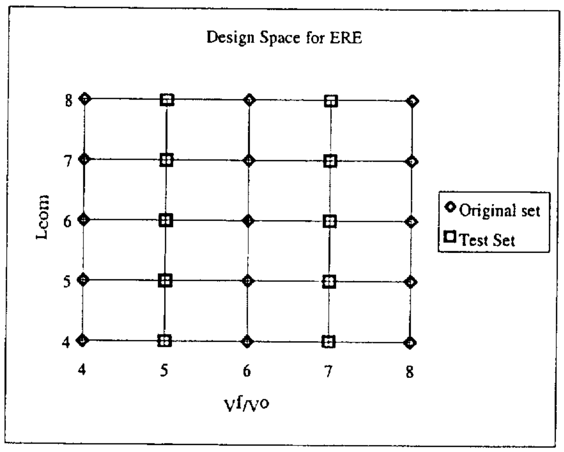
Scheme	σ for η (%)	σ for W (%)
RBNN (*Solverbe*)	1.251	1.096
RBNN (*Solverb*)	0.292	1.102
BPNN	0.777	2.563
Reduced Cubic RS	1.031	1.223

Table 8: RMS error in predicting the values of the objective function by various schemes for the supersonic turbine. (the error are given in percentages of the mean value of the responses)

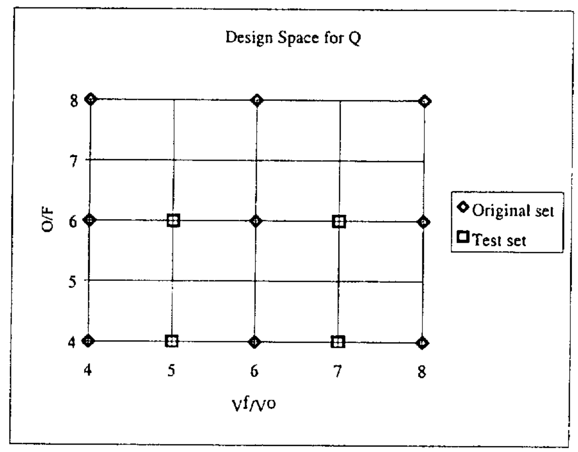
Scheme	*D*, in.	*RPM*	A_{ann}, in^2	C_v, in.	C_b, in.	K_r, %	η	*W*, lbs	V_{pitch}, in./sec	$AN^2 * 10^{10}$, $in^2 * rpm^2$	Δpay, lbs
RBNN (*Solverbe*)	9.88	37086.8	43.62	1.10	0.95	0.0	0.508 (5.80)	725.17 (0.74)	1600.0	6.00	-537.17 (29.80)

Meanline	9.88	37086.8	43.62	1.10	0.95	0.0	**0.480**	**730.61**	**1600.0**	**6.00**	**-765.22**
RBNN (*Solverb*)	10.15	36090.5	46.06	1.13	0.90	0.0	0.492 (1.75)	744.45 (0.17)	1600.0	6.00	-684.45 (9.16)
Meanline	10.15	36090.5	46.06	1.13	0.90	0.0	**0.484**	**745.76**	**1600**	**6.00**	**-753.45**
BPNN	10.41	35188.6	48.46	0.89	1.30	0.0	0.497 (2.49)	692.96 (8.63)	1600.0	6.00	-592.96 (21.49)
Meanline	10.41	35188.6	48.46	0.89	1.30	0.0	**0.484**	**758.76**	**1600.0**	**6.00**	**-755.23**
Reduced Cubic RS	9.18	39926.7	37.64	1.30	0.99	0.0	0.475 (1.50)	674.21 (2.10)	1600.0	6.00	-750.21 (8.40)
Meanline	9.18	39926.7	37.64	1.30	0.99	0.0	**0.468**	**688.44**	**1600.0**	**6.00**	**-819.00**

Table 9: Optimal Solutions with constraints on V_{rim} and AN^2 for a supersonic turbine. (error given in parenthesis for each prediction is in %)

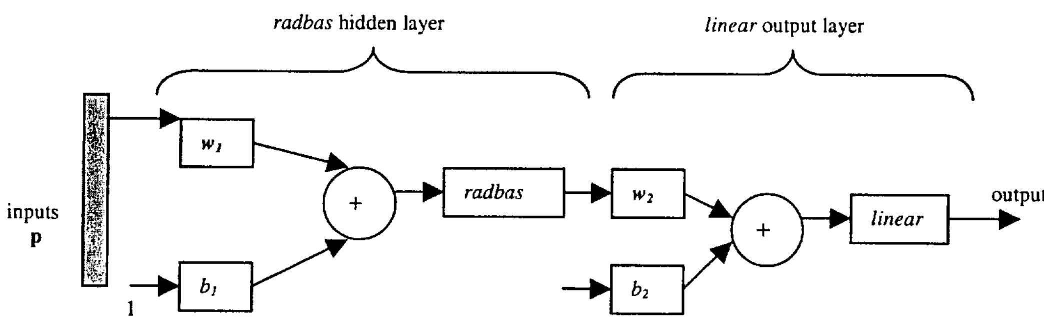


(a)



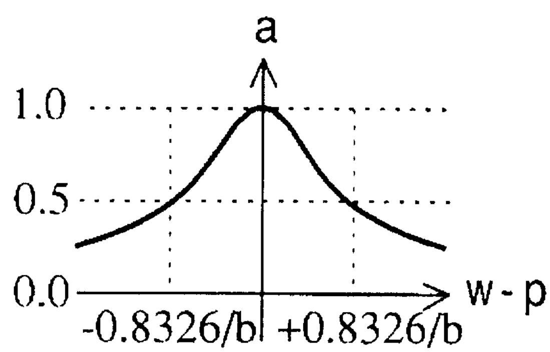
(b)

Figure 1: Design space for (a) *ERE*. (15 Training points, 10 Test points) (b) *Q*. (9 Training points, 4 Test points) for the injector.



(a)

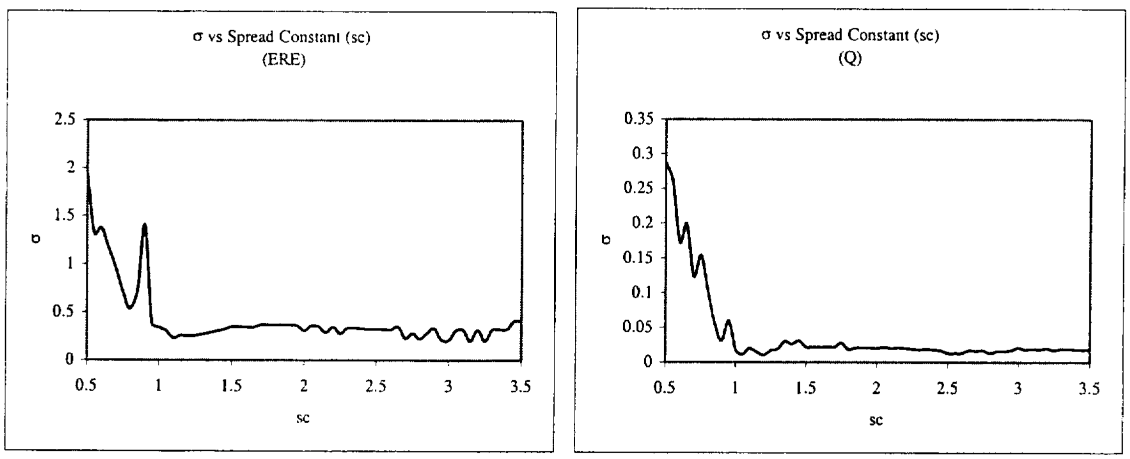
Figure 2: (a) Radial basis network, (b) Transfer function, *radbas*. (Continued)



$$a = radbas\left(dist\left(w, p\right)\times b\right)$$

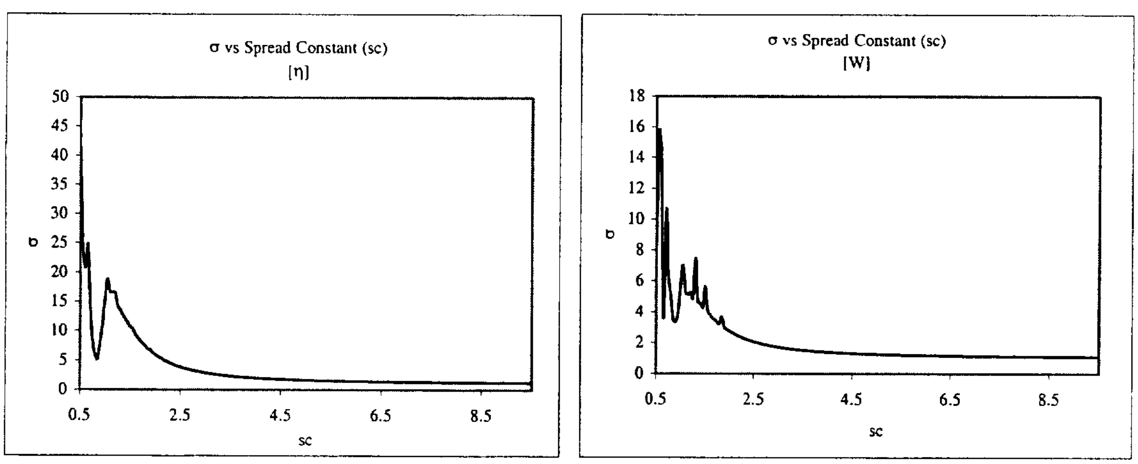
(b)

Figure 2: (a) Radial basis network, (b) Transfer function, *radbas*.



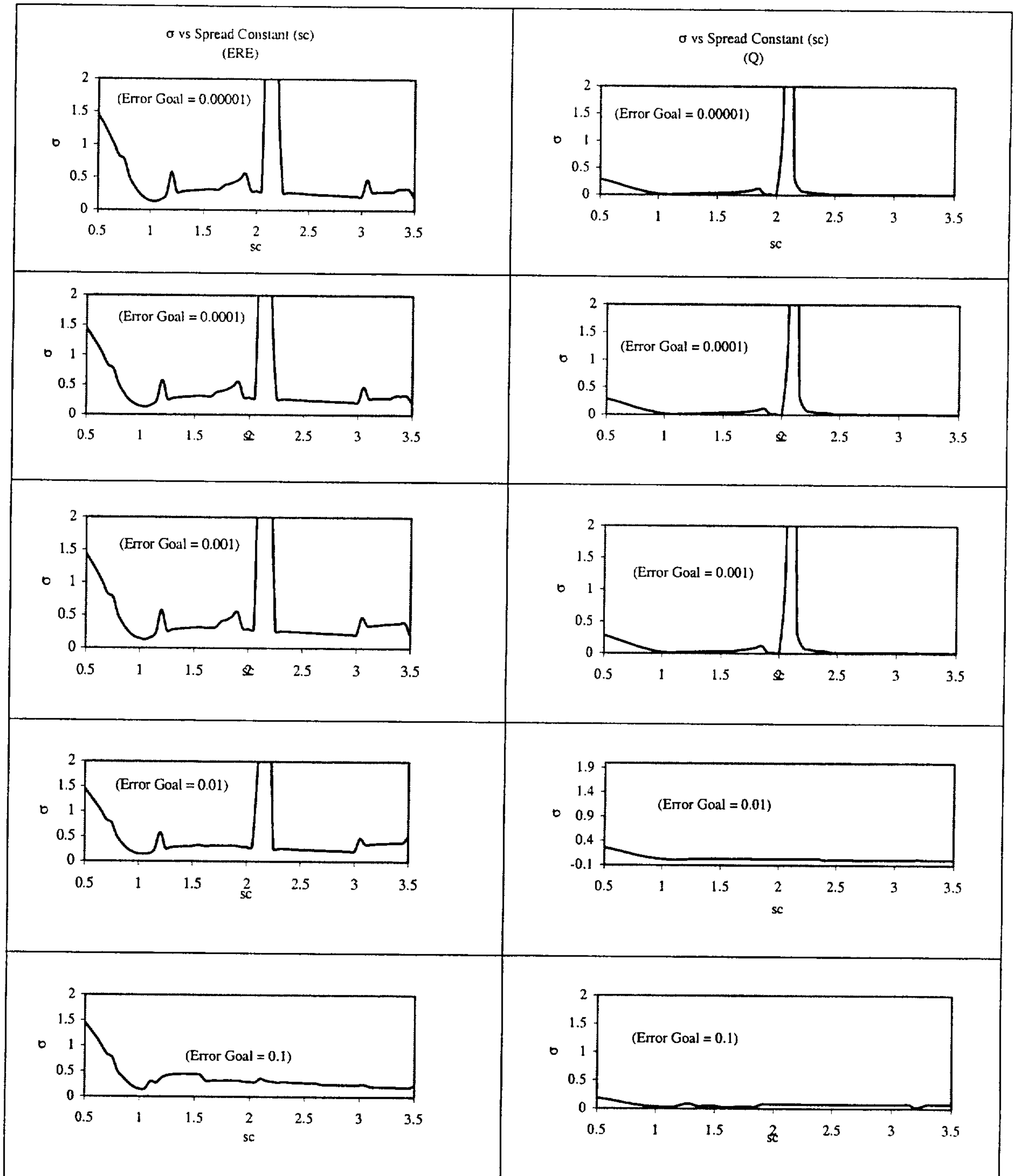
(a) (b)

Figure 3: Comparison of σ for different NN designed with *solverbe* for (a) *ERE* (%) and (b) *Q* (Btu/in^2-sec).



(a) (b)

Figure 4: Comparison of σ for different NN designed with *solverbe* for (a) η (%) and (b) *W* (lbs).



(a) (b)

Figure 5: Comparison of σ for different NN designed with *solverb* for (a) *ERE* (%) and (b) *Q* (Btu/in^2-sec).

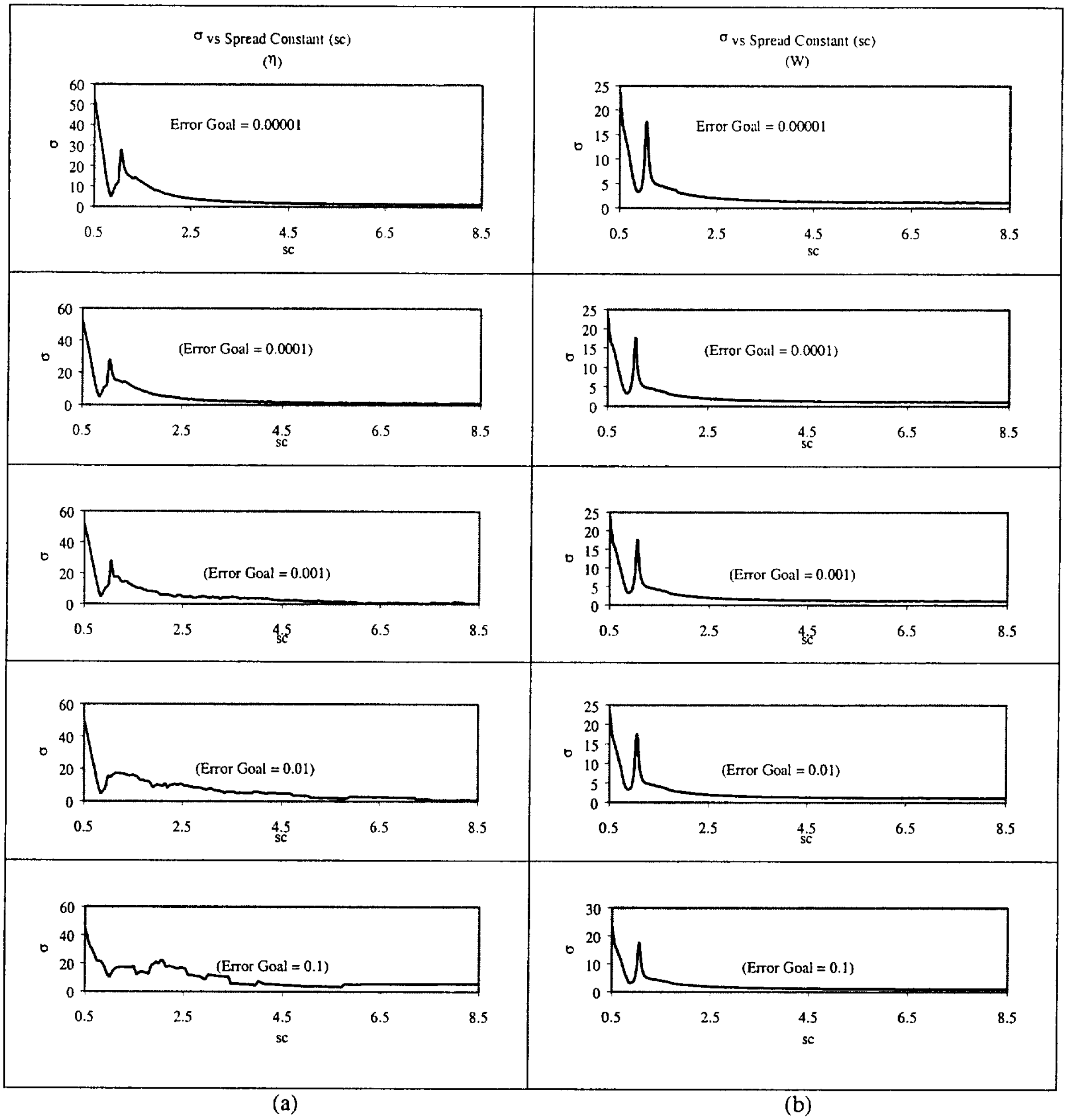
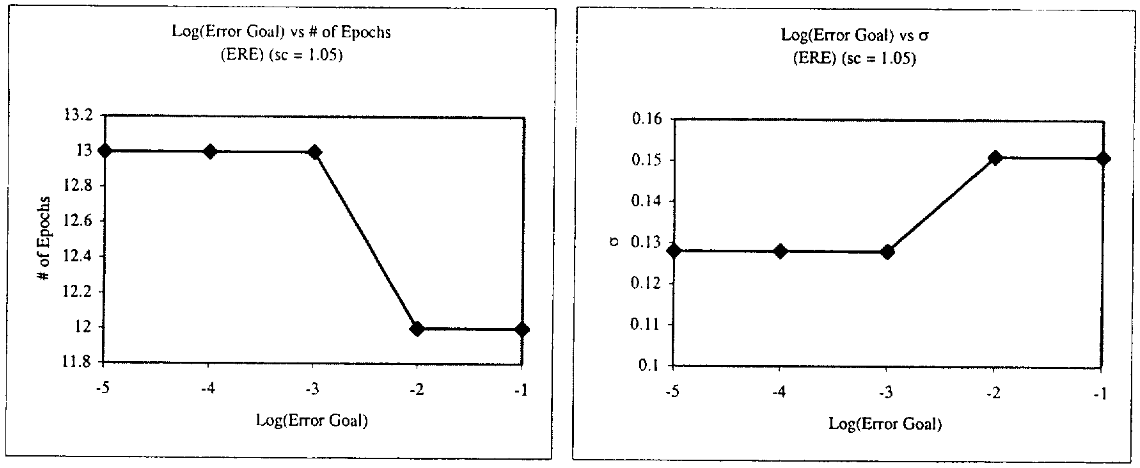
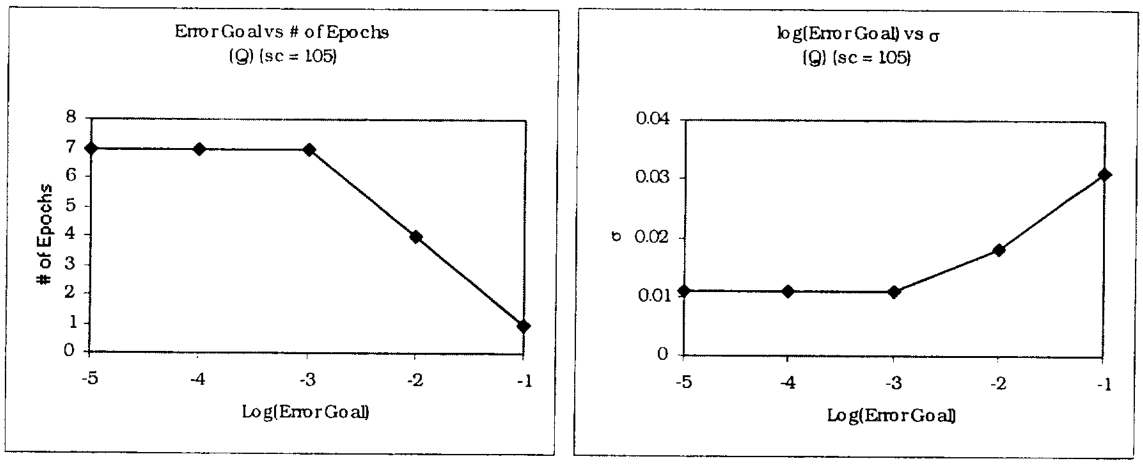


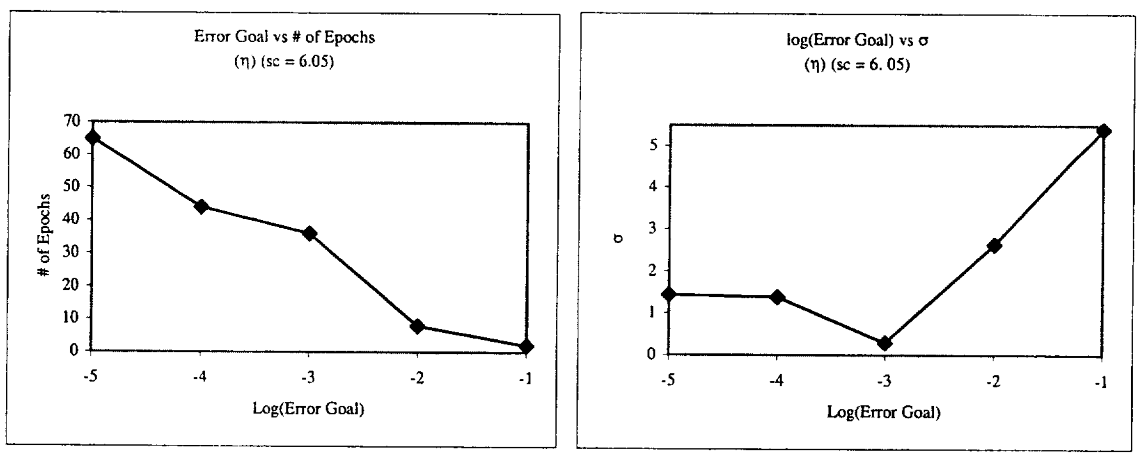
Figure 6: Comparison of σ for different NN designed with *solverb* for (a) η (%) and (b) W (lbs).



(a)

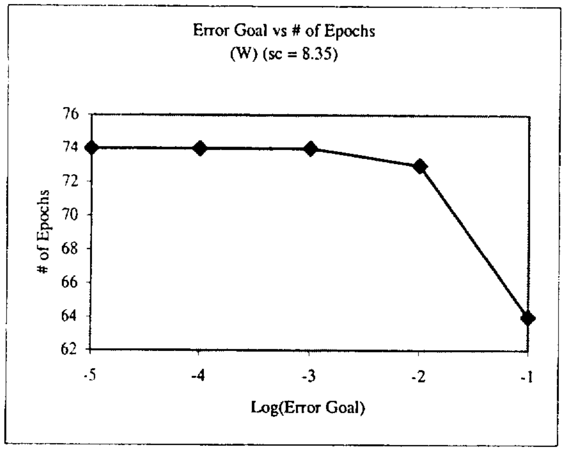


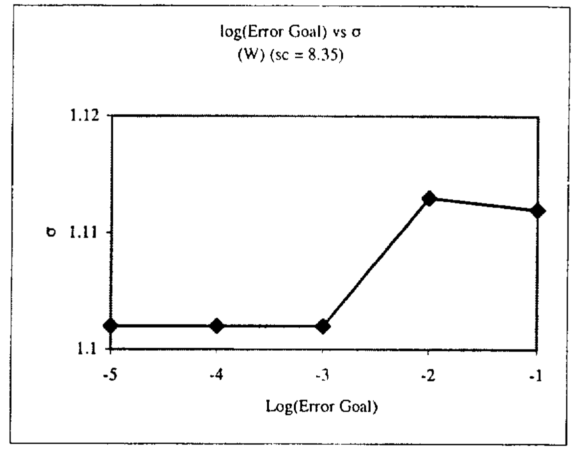
(b)



(c)

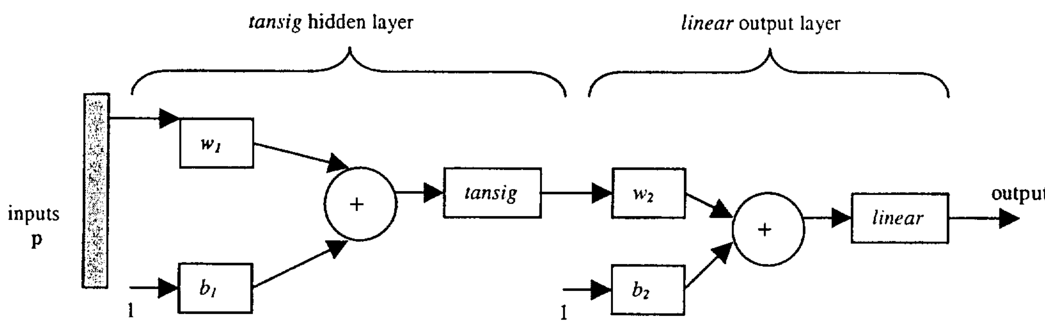
Figure 7: Comparison of *Error goal* vs number of Epochs and σ for networks trained with *solverb*. (a) *ERE* (%), (b) Q (Btu/in^2-sec), (c) η (%) and (d) W (lbs). (Continued)



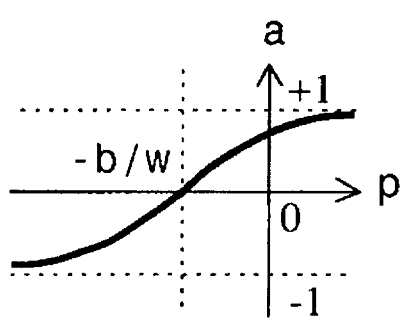


(d)

Figure 7: Comparison of *Error goal* vs number of Epochs and σ for networks trained with *solverb*. (a) *ERE* (%), (b) Q (Btu/in^2-sec), (c) η (%) and (d) W (lbs).



(a)



$$a = tansig\left(w \cdot p + b\right)$$

(b)

Figure 8: (a) Back-propagation Network, (b) Transfer function, *tansig*.

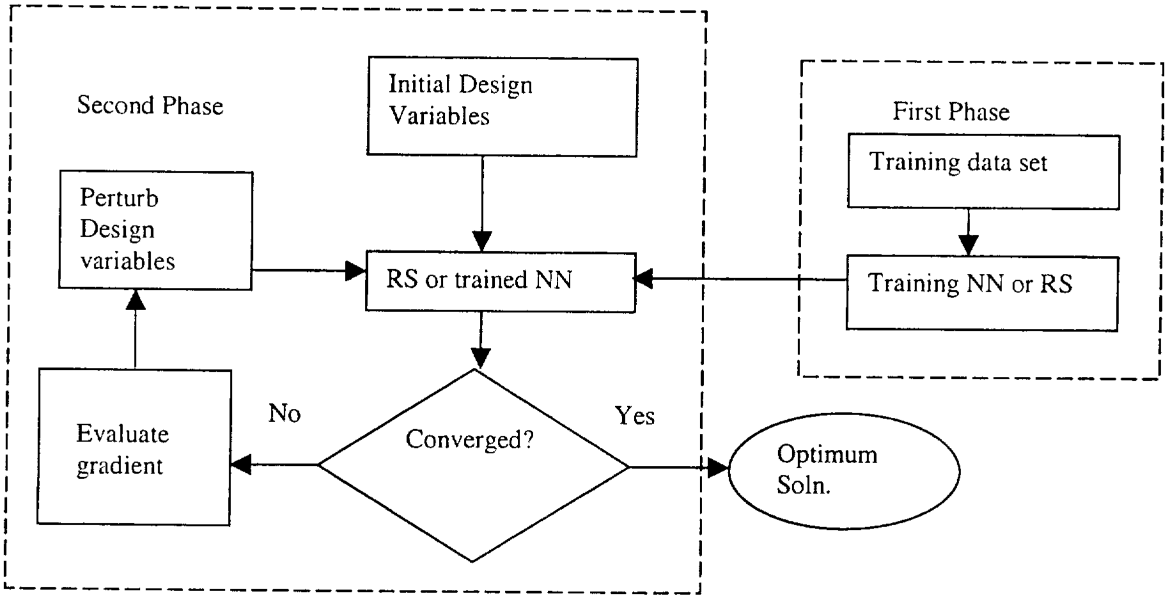
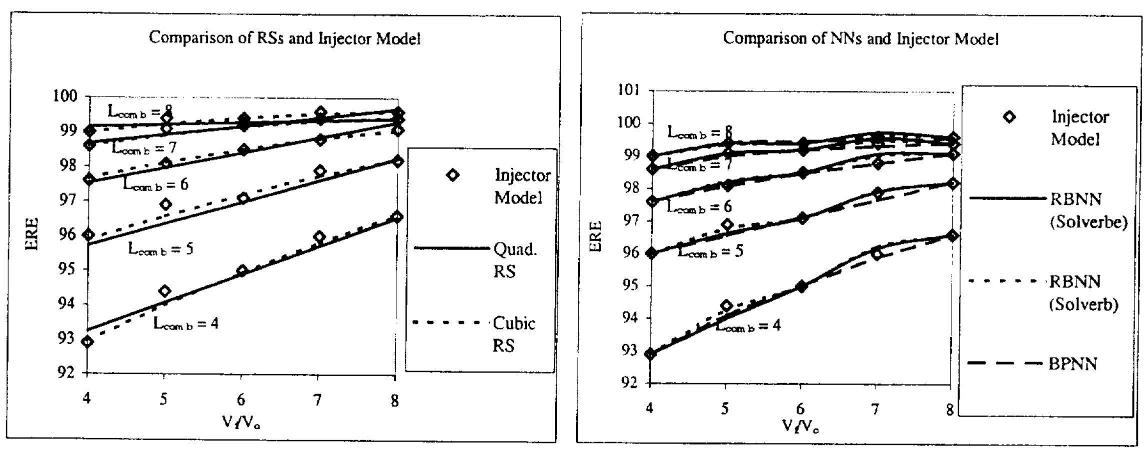
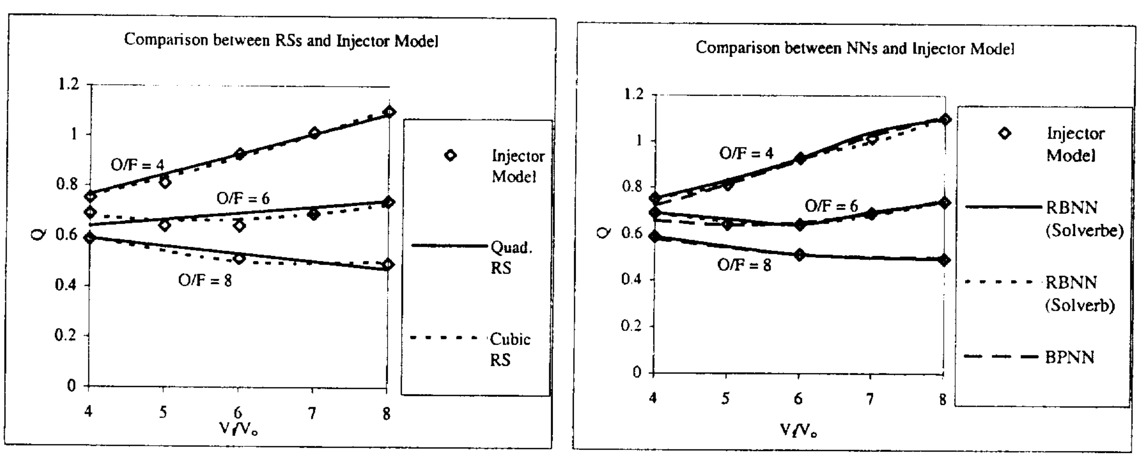


Figure 9: Schematic of the optimization process.



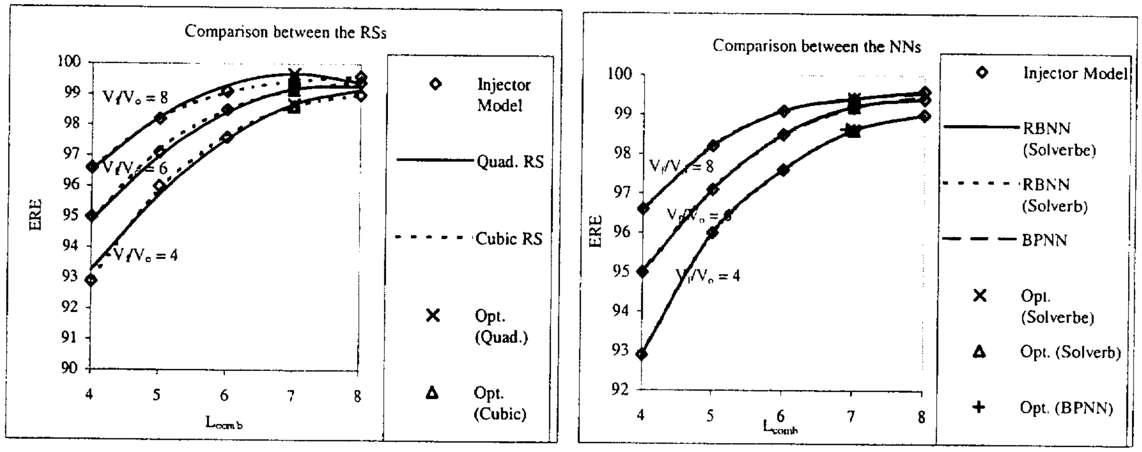
(a) (b)

Figure 10: Comparison of models with test data for ERE of the injector. (a) RSs (b) NNs.



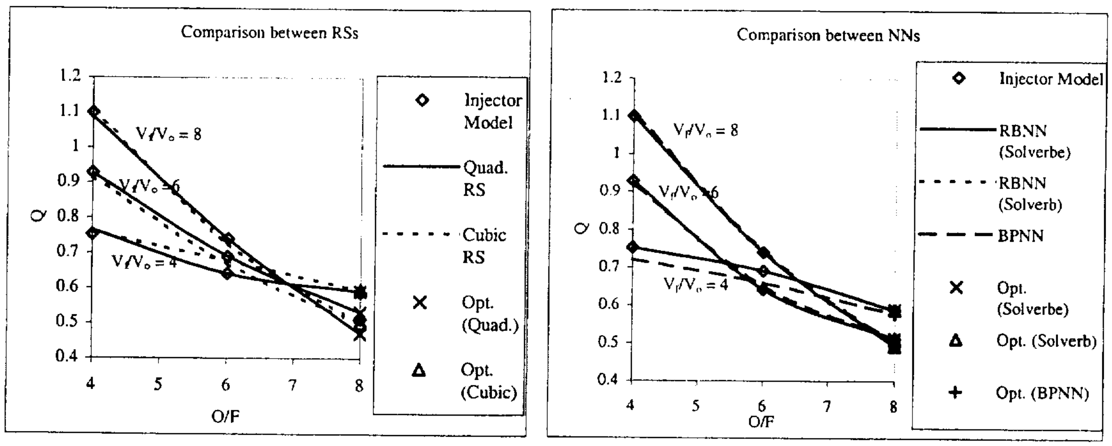
(a) (b)

Figure 11: Comparison of models with test data for Q of the injector. (a) RSs, (b) NNs.



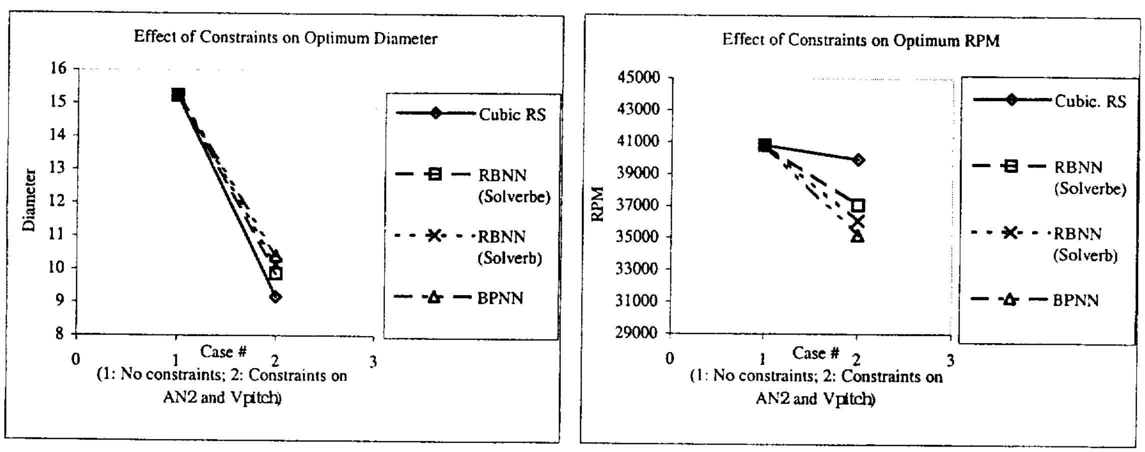
(a) (b)

Figure 12: Performance of models for ERE of the injector. (a) RSs (b) NNs.



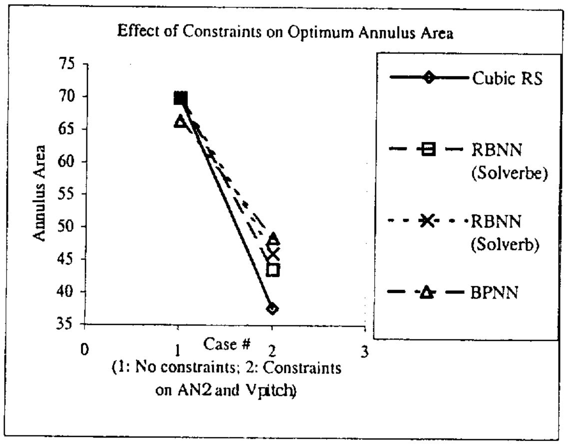
(a) (b)

Figure 13: Performance of models for Q of the injector. (a) RSs (b) NNs.

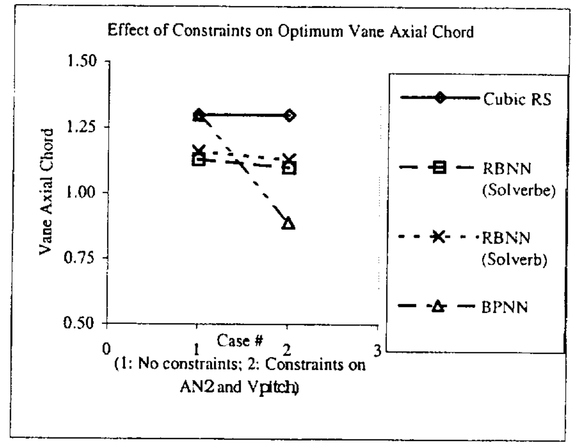


(a) (b)

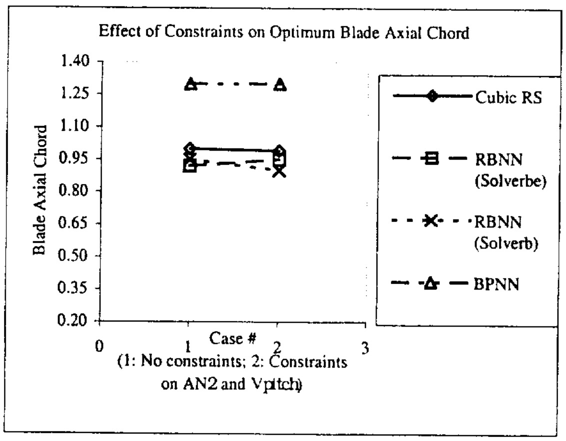
Figure 14: Effect due to presence (case 1) or lack of constraints (case 2) on design variables. (a) Optimum Diameter, D (in.), (b) Optimum RPM, (c) Optimum Annulus Area, A_{ann} (in.^{2}), (d) Optimum Vane Axial Chord, C_V (in.), (e) Optimum Blade Axial Chord, C_b (in.) and (f) Stage Reaction, K_r (%). (Continued)



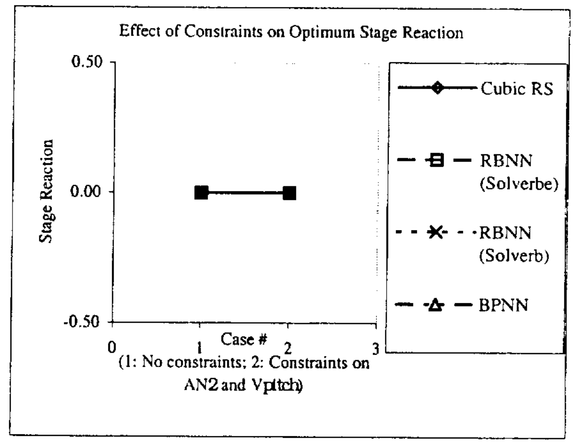
(c)



(d)

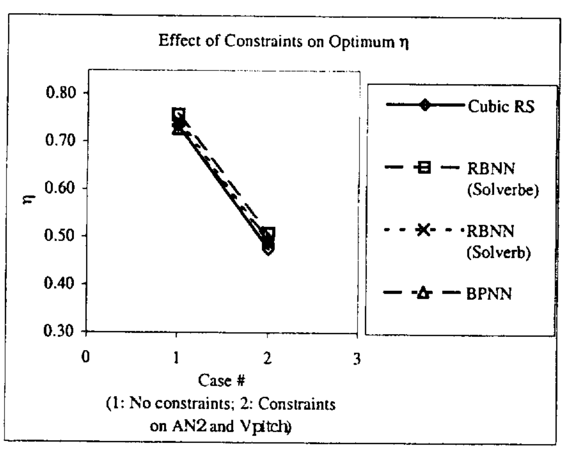


(e)

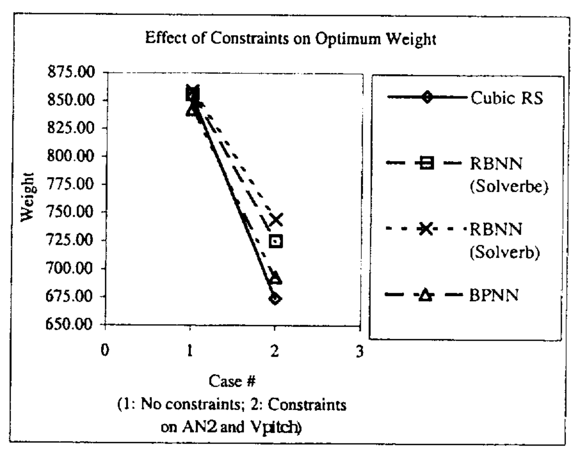


(f)

Figure 14: Effect due to presence (case 1) or lack of constraints (case 2) on design variables. (a) Optimum Diameter, D (in.), (b) Optimum RPM, (c) Optimum Annulus Area, A_{ann} (in.^{2}), (d) Optimum Vane Axial Chord, C_V (in.), (e) Optimum Blade Axial Chord, C_b (in.) and (f) Stage Reaction, K_r (%).

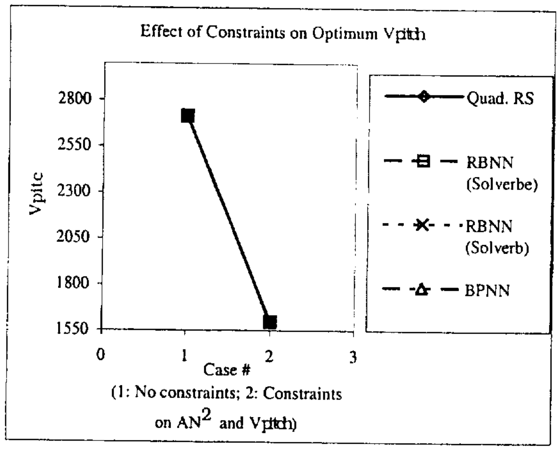


(a)

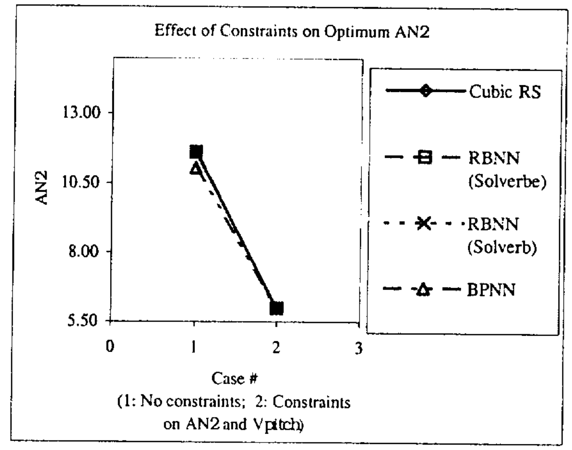


(b)

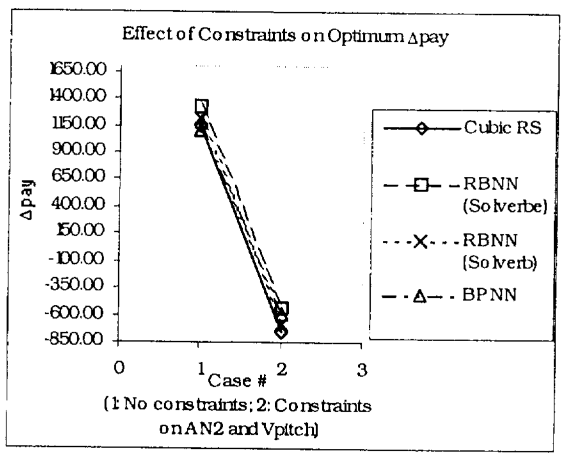
Figure 15: Effect due to presence (case 1) or lack of constraints (case 2) on objective functions. (a) Optimum Efficiency, η (%), (b) Optimum Weight, W (lbs), (c) Optimum pitch speed, V_{pitch} (in. /sec), (d) Optimum Annulus Area X RPM, AN^2 (in^2*rpm^2) and (e) Optimum Incremental Payload, Δpay (lbs). (Continued).



(c)



(d)



(e)

Figure 15: Effect due to presence (case 1) or lack of constraints (case 2) on objective functions. (a) Optimum Efficiency, η (%), (b) Optimum Weight, W (lbs), (c) Optimum pitch speed, V_{pitch} (in. /sec), (d) Optimum Annulus Area X RPM, AN^2 (in^2*rpm^2) and (e) Optimum Incremental Payload, Δpay (lbs).

APPENDIX

O/F	V_f/V_o	L_{comb}, in.	ERE, %	Q, Btu/in^2-sec
4.0	4.0	4.0	92.9	0.753
4.0	4.0	5.0	96.0	0.753
4.0	4.0	6.0	97.6	0.753
4.0	4.0	7.0	98.6	0.753
4.0	4.0	8.0	99.0	0.753
4.0	6.0	4.0	95.0	0.928
4.0	6.0	5.0	97.1	0.928
4.0	6.0	6.0	98.5	0.928
4.0	6.0	7.0	99.2	0.928
4.0	6.0	8.0	99.4	0.928
4.0	8.0	4.0	96.6	1.10
4.0	8.0	5.0	98.2	1.10
4.0	8.0	6.0	99.1	1.10
4.0	8.0	7.0	99.4	1.10
4.0	8.0	8.0	99.6	1.10

Table 1A. Performance and heat flux responses for $O/F = 4$ for the shear coaxial injector element. (Tables 1A-3A together contain 45 data points used as the training set)

O/F	V_f/V_o	L_{comb}, in.	ERE, %	Q, Btu/in^2-sec
6.0	4.0	4.0	92.9	0.691
6.0	4.0	5.0	96.0	0.691
6.0	4.0	6.0	97.6	0.691
6.0	4.0	7.0	98.6	0.691
6.0	4.0	8.0	99.0	0.691
6.0	6.0	4.0	95.0	0.642
6.0	6.0	5.0	97.1	0.642
6.0	6.0	6.0	98.5	0.642
6.0	6.0	7.0	99.2	0.642
6.0	6.0	8.0	99.4	0.642
6.0	8.0	4.0	96.6	0.741
6.0	8.0	5.0	98.2	0.741
6.0	8.0	6.0	99.1	0.741
6.0	8.0	7.0	99.4	0.741
6.0	8.0	8.0	99.6	0.741

Table 2A. Performance and heat flux responses for $O/F = 6$ for the shear coaxial injector element.

O/F	V_f/V_o	L_{comb}, in.	*ERE*, %	*Q*, Btu/in^2-sec
8.0	4.0	4.0	92.9	0.588
8.0	4.0	5.0	96.0	0.588
8.0	4.0	6.0	97.6	0.588
8.0	4.0	7.0	98.6	0.588
8.0	4.0	8.0	99.0	0.588
8.0	6.0	4.0	95.0	0.512
8.0	6.0	5.0	97.1	0.512
8.0	6.0	6.0	98.5	0.512
8.0	6.0	7.0	99.2	0.512
8.0	6.0	8.0	99.4	0.512
8.0	8.0	4.0	96.6	0.493
8.0	8.0	5.0	98.2	0.493
8.0	8.0	6.0	99.1	0.493
8.0	8.0	7.0	99.4	0.493
8.0	8.0	8.0	99.6	0.493

Table 3A. Performance and heat flux responses for *O/F* = 8 for the shear coaxial injector element.

O/F	V_f/V_o	L_{comb}, in.	*ERE*, %	*Q*, Btu/in^2-sec
4.0	5.0	4.0	94.4	0.812
4.0	5.0	5.0	96.9	0.812
4.0	5.0	6.0	98.1	0.812
4.0	5.0	7.0	99.1	0.812
4.0	5.0	8.0	99.4	0.812
4.0	7.0	4.0	96.0	1.014
4.0	7.0	5.0	97.9	1.014
4.0	7.0	6.0	98.8	1.014
4.0	7.0	7.0	99.4	1.014
4.0	7.0	8.0	99.6	1.014
6.0	5.0	4.0	94.4	0.642
6.0	5.0	5.0	96.9	0.642
6.0	5.0	6.0	98.1	0.642
6.0	5.0	7.0	99.1	0.642
6.0	5.0	8.0	99.4	0.642
6.0	7.0	4.0	96.0	0.691
6.0	7.0	5.0	97.9	0.691
6.0	7.0	6.0	98.8	0.691
6.0	7.0	7.0	99.4	0.691
6.0	7.0	8.0	99.6	0.691

Table 4A: Data used to test the RS and NN for the shear coaxial injector element. (The table contains 20 data points used as the testing set)

Mean Diameter, D (in.)	RPM	Blade Annulus Area, A_{ann} (in^2)	Vane Axial Chord, C_v (in.)	Blade Axial Chord, C_b (in.)	Stage Reaction, K_r (%)	η (t-s)	Weight, W (lbs)	Pitch Speed, V_{pitch} (in./sec)	$AN^2 * 10^{10}$ (in^2* rpm^2)
5.081	18837.6	37.6379	0.3	0.3	0.0	0.1303	501.45	417.9644	1.34
5.081	18837.6	37.6379	0.3	0.3	0.0	0.1188	474.55	417.9644	1.34
5.081	18837.6	37.6379	0.3	0.3	0.5	0.1364	515.59	417.9644	1.34
5.081	18837.6	37.6379	0.3	0.3	0.5	0.1220	482.19	417.9644	1.34
5.081	18837.6	37.6379	1.3	1.3	0	0.1371	517.05	417.9644	1.34
5.081	18837.6	37.6379	1.3	1.3	0	0.1214	480.68	417.9644	1.34
5.081	18837.6	37.6379	1.3	1.3	0.5	0.1444	533.4	417.9644	1.34
5.081	18837.6	37.6379	1.3	1.3	0.5	0.1247	488.51	417.9644	1.34
5.081	18837.6	69.8989	0.3	0.3	0	0.1421	528.29	417.9644	2.48
5.081	18837.6	69.8989	0.3	0.3	0	0.1233	485.3	417.9644	2.48
5.081	18837.6	69.8989	0.3	0.3	0.5	0.1474	540.05	417.9644	2.48
5.081	18837.6	69.8989	0.3	0.3	0.5	0.1265	492.63	417.9644	2.48
5.081	18837.6	69.8989	1.3	1.3	0	0.1483	542.03	417.9644	2.48
5.081	18837.6	69.8989	1.3	1.3	0	0.1259	491.39	417.9644	2.48
5.081	18837.6	69.8989	1.3	1.3	0.5	0.1545	555.51	417.9644	2.48
5.081	18837.6	69.8989	1.3	1.3	0.5	0.1291	498.88	417.9644	2.48
5.081	43954.4	37.6379	0.3	0.3	0	0.2842	481.62	975.2502	7.27
5.081	43954.4	37.6379	0.3	0.3	0	0.2587	455.22	975.2502	7.27
5.081	43954.4	37.6379	0.3	0.3	0.5	0.2653	462.14	975.2502	7.27
5.081	43954.4	37.6379	1.3	1.3	0	0.3012	498.79	975.2502	7.27
5.081	43954.4	37.6379	1.3	1.3	0	0.2644	461.25	975.2502	7.27
5.081	43954.4	37.6379	1.3	1.3	0.5	0.3155	512.86	975.2502	7.27
5.081	43954.4	37.6379	1.3	1.3	0.5	0.2713	468.41	975.2502	7.27
5.081	43954.4	69.8989	0.3	0.3	0	0.3107	508.13	975.2502	13.5
5.081	43954.4	69.8989	0.3	0.3	0	0.2692	466.29	975.2502	13.5
5.081	43954.4	69.8989	0.3	0.3	0.5	0.3214	518.52	975.2502	13.5
5.081	43954.4	69.8989	0.3	0.3	0.5	0.2758	473.1	975.2502	13.5
5.081	43954.4	69.8989	1.3	1.3	0	0.3261	523.11	975.2502	13.5
5.081	43954.4	69.8989	1.3	1.3	0	0.275	472.29	975.2502	13.5
5.081	43954.4	69.8989	1.3	1.3	0.5	0.3384	534.87	975.2502	13.5
5.081	43954.4	69.8989	1.3	1.3	0.5	0.2819	479.28	975.2502	13.5
15.243	18837.6	37.6379	0.3	0.3	0	0.3425	895.71	1253.893	1.34
15.243	18837.6	37.6379	0.3	0.3	0	0.3078	840.05	1253.893	1.34
15.243	18837.6	37.6379	0.3	0.3	0.5	0.3495	906.58	1253.893	1.34
15.243	18837.6	37.6379	0.3	0.3	0.5	0.3093	842.6	1253.893	1.34
15.243	18837.6	37.6379	1.3	1.3	0	0.3604	923.49	1253.893	1.34
15.243	18837.6	37.6379	1.3	1.3	0	0.3083	840.98	1253.893	1.34
15.243	18837.6	37.6379	1.3	1.3	0.5	0.3688	936.31	1253.893	1.34
15.243	18837.6	37.6379	1.3	1.3	0.5	0.3099	843.58	1253.893	1.34
15.243	18837.6	69.8989	0.3	0.3	0	0.3744	944.92	1253.893	2.48
15.243	18837.6	69.8989	0.3	0.3	0	0.3204	860.49	1253.893	2.48

Table 5A: Data used to generate the RS and train the NN for the supersonic turbine. (Continued)

Mean Diameter, D (in.)	RPM	Blade Annulus Area, A_{ann} (in^2)	Vane Axial Chord, C_v (in.)	Blade Axial Chord, C_b (in.)	Stage Reaction, K_r (%)	η (t-s)	Weight, W (lbs)	Pitch Speed, V_{pitch} (in./sec)	AN^2 * 10^{10} (in^2* rpm^2)
15.243	18837.6	69.8989	0.3	0.3	0.5	0.3832	958.17	1253.893	2.48
15.243	18837.6	69.8989	0.3	0.3	0.5	0.3243	866.9	1253.893	2.48
15.243	18837.6	69.8989	1.3	1.3	0	0.3905	968.95	1253.893	2.48
15.243	18837.6	69.8989	1.3	1.3	0	0.3219	862.9	1253.893	2.48
15.243	18837.6	69.8989	1.3	1.3	0.5	0.4003	983.59	1253.893	2.48
15.243	18837.6	69.8989	1.3	1.3	0.5	0.3259	869.38	1253.893	2.48
15.243	43954.4	37.6379	0.3	0.3	0	0.608	760.14	2925.751	7.27
15.243	43954.4	37.6379	0.3	0.3	0	0.5751	735.2	2925.751	7.27
15.243	43954.4	37.6379	0.3	0.3	0.5	0.6156	765.89	2925.751	7.27
15.243	43954.4	37.6379	0.3	0.3	0.5	0.5713	732.32	2925.751	7.27
15.243	43954.4	37.6379	1.3	1.3	0	0.657	796.37	2925.751	7.27
15.243	43954.4	37.6379	1.3	1.3	0	0.5762	736.08	2925.751	7.27
15.243	43954.4	37.6379	1.3	1.3	0.5	0.6666	803.33	2925.751	7.27
15.243	43954.4	37.6379	1.3	1.3	0.5	0.5723	733.11	2925.751	7.27
15.243	43954.4	69.8989	0.3	0.3	0	0.6754	809.67	2925.751	13.5
15.243	43954.4	69.8989	0.3	0.3	0	0.6035	756.8	2925.751	13.5
15.243	43954.4	69.8989	0.3	0.3	0.5	0.6888	819.3	2925.751	13.5
15.243	43954.4	69.8989	0.3	0.3	0.5	0.6052	758.09	2925.751	13.5
15.243	43954.4	69.8989	1.3	1.3	0	0.7202	841.52	2925.751	13.5
15.243	43954.4	69.8989	1.3	1.3	0	0.6064	759	2925.751	13.5
15.243	43954.4	69.8989	1.3	1.3	0.5	0.7348	851.67	2925.751	13.5
15.243	43954.4	69.8989	1.3	1.3	0.5	0.6084	760.47	2925.751	13.5
5.081	31396	53.7684	0.8	0.8	0.25	0.2285	517.15	696.6073	5.3
15.243	31396	53.7684	0.8	0.8	0.25	0.5412	867.47	2089.822	5.3
10.162	18837.6	53.7684	0.8	0.8	0.25	0.2636	765.49	835.9288	1.91
10.162	43954.4	53.7684	0.8	0.8	0.25	0.5319	701.6	1950.5	10.4
10.162	31396	37.6379	0.8	0.8	0.25	0.3966	719.85	1393.215	3.71
10.162	31396	69.8989	0.8	0.8	0.25	0.4139	738.52	1393.215	6.89
10.162	31396	53.7684	0.3	0.3	0.25	0.3988	722.24	1393.215	5.3
10.162	31396	53.7684	1.3	1.3	0.25	0.4087	732.98	1393.215	5.3
10.162	31396	53.7684	0.8	0.8	0	0.4007	724.3	1393.215	5.3
10.162	31396	53.7684	0.8	0.8	0.5	0.4093	733.6	1393.215	5.3
10.162	31396	53.7684	0.8	0.8	0.25	0.4382	764.27	1393.215	5.3
10.162	31396	53.7684	0.8	0.8	0.25	0.3687	689.01	1393.215	5.3
10.162	31396	53.7684	0.8	0.8	0.25	0.4094	733.74	1393.215	5.3

Table 5A: Data used to generate the RS and train the NN for the supersonic turbine. (The table contains 76 data points used as the training set)

Mean Diameter, D (in.)	RPM	Blade Annulus Area, A_{ann} (in^2)	Vane Axial Chord, C_v (in.)	Blade Axial Chord, C_b (in.)	Stage Reaction, K_r (%)	η (t-s)	Weight, W (lbs)	Pitch Speed, V_{pitch} (in./sec)	$AN^2 * 10^{10}$ ($in^2 * rpm^2$)
5.5891	20093.44	39.251	0.35	0.35	0.03	0.1582	541.999	490.41	1.58
6.0972	21349.28	40.864	0.4	0.4	0.05	0.1863	576.690	568.43	1.86
6.6053	22605.12	42.477	0.45	0.45	0.08	0.2147	606.644	652.02	2.17
7.1134	23860.96	44.090	0.5	0.5	0.10	0.2432	632.936	741.19	2.51
7.6215	25116.8	45.703	0.55	0.55	0.13	0.2717	655.980	835.93	2.88
8.1296	26372.64	47.316	0.6	0.6	0.15	0.3001	676.217	936.24	3.29
8.6377	27628.48	48.929	0.65	0.65	0.18	0.3283	693.910	1042.12	3.73
9.1458	28884.32	50.542	0.7	0.7	0.20	0.3560	709.287	1153.58	4.22
9.6539	30140.16	52.155	0.75	0.75	0.23	0.3830	722.487	1270.61	4.74
10.6701	32651.84	55.381	0.85	0.85	0.28	0.4350	743.202	1521.39	5.90
11.1782	33907.68	56.995	0.9	0.9	0.30	0.4597	751.038	1655.14	6.55
11.6863	35163.52	58.608	0.95	0.95	0.33	0.4833	757.273	1794.46	7.25
12.1944	36419.36	60.221	1	1	0.35	0.5058	761.977	1939.35	7.99
12.7025	37675.2	61.834	1.05	1.05	0.38	0.5269	765.201	2089.82	8.78
13.2106	38931.04	63.447	1.1	1.1	0.40	0.5466	766.977	2245.86	9.62
13.7187	40186.88	65.060	1.15	1.15	0.43	0.5646	767.347	2407.47	10.51
14.2268	41442.72	66.673	1.2	1.2	0.45	0.5810	766.329	2574.66	11.45
14.7349	42698.56	68.286	1.25	1.25	0.48	0.5955	763.961	2747.42	12.45

Table 6A: Data used to test the RS and NN for the supersonic turbine. (The table contains 18 data points used as the testing set)

JOURNAL OF PROPULSION AND POWER
Vol. 17, No. 2, March–April 2001

Response Surface and Neural Network Techniques for Rocket Engine Injector Optimization

Wei Shyy*
University of Florida, Gainesville, Florida 32611-6250
P. Kevin Tucker†
NASA Marshall Space Flight Center, Alabama 35812
and
Rajkumar Vaidyanathan‡
University of Florida, Gainesville, Florida 32611-6250

The response surface methodology for rocket engine injector design optimization for which only modest amounts of data may exist is examined. Two main aspects are emphasized: relative performance of quadratic and cubic polynomial response surfaces and enhancement of the fidelity of the response surface via neural networks. A data set of 45 design points from a semi-empirical model for a shear coaxial injector element using gaseous oxygen and gaseous hydrogen propellants is used to formulate response surfaces using quadratic and cubic polynomials. This original data set is also employed to train a two-layered radial basis neural network (RBNN). The trained network is then used to generate additional data to augment the original information available to characterize the design space. Quadratic and cubic polynomials are again used to generate response surfaces for this RBNN-enhanced data set. The response surfaces resulting from both the original and RBNN-enhanced data sets are compared for accuracy. Whereas the cubic fit is superior to the quadratic fit for each data set, the RBNN-enhanced data set is capable of improving the accuracy of the response surface if noticeable errors from polynomial curve fits are encountered. Furthermore, the RBNN-enhanced data set yields more consistent selections of optimal designs between cubic and quadratic polynomials. The techniques developed can be directly applied to injector design and optimization for rocket propulsion.

Nomenclature

A	=	lowest acceptable value of energy release efficiency (ERE)
a	=	radial basis neural network output
B	=	target value of ERE
b	=	bias associated with a neuron in neural networks
C	=	target value of Q
D	=	composite desirability function
d_1	=	desirability function related to ERE
d_2	=	desirability function related to Q
E	=	highest acceptable value of Q
e_i	=	error at the ith design point
L_{comb}	=	combustor length (length from injector to throat)
n	=	number of data points
n_p	=	number of coefficients in the response surface
O/F	=	propellant mixture ratio
p	=	input vector of the neural network
Q	=	actual chamber wall heat flux
Q_{nom}	=	nominal chamber wall heat flux
radbas	=	transfer function of radial basis neural network
s	=	weighting factor for d_1
t	=	weighting factor for d_2
σ	=	root mean square error
σ_a	=	adjusted root mean square error

Received 15 June 1999; revision received 10 August 2000; accepted for publication 13 August 2000.

*Professor, Department Chair, Department of Aerospace Engineering, Mechanics, and Engineering Science.

†Aerospace Technologist, Space Transportation Directorate, NASA Marshall Space Flight Center.

‡Graduate Student, Department of Aerospace Engineering, Mechanics, and Engineering Science.

I. Introduction

THE injector design methodologies used successfully in previous rocket propulsion system development programs were typically based on large subscale databases and the empirical design tools derived from them.[1–5] Extensive sub- and full-scale hot-fire test programs often guided these methodologies. Current and planned launch vehicle programs have tight budgets and aggressive schedules, neither of which is conductive to the large test programs of the past. Also, new requirements for operability and maintainability require that injector design be robust. Hence, variables not previously included in the injector design now merit consideration for inclusion in the design process. Also, the effect of the injector design on variables, peripheral to, but influenced by the injector, may need to be included in the injector design process. These new programs with compressed schedules, lower budgets, and more stringent requirements make development of broader and more efficient injector design methodologies an attractive goal.

Historically, injectors have been designed, fabricated, and tested based on experience and intuition. As hardware was tested, designers proposed modifications aimed at obtaining an improved design. Despite their experience and skill, these efforts were unlikely to produce the optimal design in a short time frame. Also, as more design variables are considered, the design process becomes increasingly complex, and it is more difficult to foresee the effect of the modification of one variable on other variables. Use of an optimization approach to guide the design addresses both of these issues. The optimization scheme allows complex, interrelated information to be managed in such a way that the extent to which variables influence each other can be objectively evaluated and optimal design points can be identified with confidence.

Development of an optimization scheme for injector design called methodology for optimizing the design of injectors (*method I*) has been reported by Tucker et al.[6] *Method I* is used to generate appropriate injector design data and then guide the designer toward an optimum design subject to the specified constraints. As reported, *method I* uses the response surface methodology (RSM) to facilitate the optimization. The RSM approach is to conduct a series of well-chosen experiments (empirical, numerical, physical, or some combination of the three) and use the resulting information to construct a global approximation (response surface) of the measured quantity (response) over the design space. A standard constrained optimization algorithm is then used to interrogate the response surface for an optimum design.

The initial demonstration of *method I* by Tucker et al.[6] focused on a simple optimization of a shear coaxial injector element with gaseous oxygen and gaseous hydrogen propellants. The design data were generated using an empirical design methodology developed by Calhoon et al.[7] These researchers conducted a large number of cold-flow and hot-fire tests over a range of propellant mixture ratios, propellant velocity ratios, and chamber pressure for shear coaxial, swirl coaxial, impinging, and premixed elements. The data were correlated directly with injector/chamber design parameters, which are recognized from both theoretical and empirical standpoints as the controlling variables. For the shear coaxial element, performance, as measured by energy release efficiency (*ERE*), is obtained using correlations taking into account combustor length L_{comb} (length from injector to throat) and the propellant velocity ratio V_f/V_0. The nominal chamber wall heat flux at a point just downstream of the injector, Q_{nom}, is calculated using a modified Bartz equation and is correlated with propellant mixture ratio O/F and propellant velocity ratio V_f/V_0 to yield the actual chamber wall heat flux Q. The objective in the initial demonstration of *method I* was to maximize injector performance while minimizing chamber wall heat flux (lower heat fluxes reduce cooling requirements and increase chamber life) and chamber length (shorter chambers lower engine weight).

The initial demonstration of *method I* used quadratic polynomials to generate the response surfaces. The surfaces for *ERE* and Q were joined by use of a desirability function, and optimum design points were sought as the independent variables (O/F, V_f/V_0, and L_{comb}) were constrained over different ranges. The initial demonstration reported by Tucker et al.[6] is viewed as a proof of concept for *method I*.

II. Scope of Current Research

Empirical design methodologies, such as that by Calhoon et al.,[7] may allow the designer to generate large quantities of data within a design space. However, due to their empiricism, these methodologies are often sufficiently accurate only over the range of variables for which test data were taken to develop the methodology. For some injector types, propellant combinations, or design conditions, this limitation may require that additional data be generated to ensure confidence in the design. Historically, these data have been generated in sub- and full-scale test programs. More recently computational fluid dynamics (CFD) analysis from validated models has been used to augment the test data. Data from test programs and CFD analysis are expensive and time consuming to obtain. Recognition of this has direct implications for the usefulness of optimization techniques in injector design methodologies. Although the optimization scheme must be capable of efficiently organizing large amounts of design information generated from empirical design methodologies, it must also be able to make effective use of the relatively small amounts of data available in some cases. An optimization scheme that requires large amounts of data to generate meaningful results will be marginally useful, if at all, when only small amounts of data are available for use.

The present effort seeks to investigate approaches that would make RSM robust and reliable for injector design optimization, especially when only limited amounts of design data exist. We first investigate the relative performance of a quadratic and a cubic polynomial for constructing response surfaces. The original data set from Tucker et al.[6] (with 45 design points) is used to generate the response surfaces for *ERE* and Q. The quality of each fit on the original data set is evaluated. Then, an approach to train a radial basis neural network (RBNN) to enhance the information available to construct the response surface is presented. Specific issues relative to the network training are evaluated and discussed. This trained RBNN is then used to generate additional design data. The data generated from the network are combined with the original data from the Calhoon et al.[7] model to form an enhanced data set, which is then refit with quadratic and cubic polynomial surfaces. The quality of the fit of the resulting surfaces is compared. Also, each surface is used to conduct design optimization over the same range of independent variables. The optimal design points are compared with exact points calculated from the empirical model.

III. Approaches

A. General

The range of propellant mixture ratios O/F propellant velocity ratios V_f/V_0, and chamber lengths L_{comb} considered in this study are shown in Table 1. Tables 2–4 shows the empirically derived performance and heat flux for the 45 combinations of O/F, V_f/V_0, and L_{comb} considered. Hereafter, these 45 design points are referred to as the original data set. As noted earlier, this original data set is augmented with additional design points generated using a trained RBNN. This new and larger data set is referred to as the enhanced data set. The RSM, using both quadratic and cubic polynomials, is used to fit both surfaces. The following two sections give pertinent details on the RSM and neural networks (NN).

Table 1 Range of design variables considered

O/F	V_f/V_0	L_{comb}, in.
4, 6, 8	4	4–8
4, 6, 8	6	4–8
4, 6, 8	8	4–8

Table 2 Performance and heat flux responses for O/F = 4 elements

O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	Q, Btu/in.^{2}-s
4.0	4.0	4.0	92.9	0.753
4.0	4.0	5.0	96.0	0.753
4.0	4.0	6.0	97.6	0.753
4.0	4.0	7.0	98.6	0.753
4.0	4.0	8.0	99.0	0.753
4.0	6.0	4.0	95.0	0.928
4.0	6.0	5.0	97.1	0.928
4.0	6.0	6.0	98.5	0.928
4.0	6.0	7.0	99.2	0.928
4.0	6.0	8.0	99.4	0.928
4.0	8.0	4.0	96.6	1.10
4.0	8.0	5.0	98.2	1.10
4.0	8.0	6.0	99.1	1.10
4.0	8.0	7.0	99.4	1.10
4.0	8.0	8.0	99.6	1.10

Table 3 Performance and heat flux responses for O/F = 6 elements

O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	Q, Btu/in.^{2}-s
6.0	4.0	4.0	92.9	0.691
6.0	4.0	5.0	96.0	0.691
6.0	4.0	6.0	97.6	0.691
6.0	4.0	7.0	98.6	0.691
6.0	4.0	8.0	99.0	0.691
6.0	6.0	4.0	95.0	0.642
6.0	6.0	5.0	97.1	0.642
6.0	6.0	6.0	98.5	0.642
6.0	6.0	7.0	99.2	0.642
6.0	6.0	8.0	99.4	0.642
6.0	8.0	4.0	96.6	0.741
6.0	8.0	5.0	98.2	0.741
6.0	8.0	6.0	99.1	0.741
6.0	8.0	7.0	99.4	0.741
6.0	8.0	8.0	99.6	0.741

Table 4 Performance and heat flux responses for O/F = 8 elements

O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s
8.0	4.0	4.0	92.9	0.588
8.0	4.0	5.0	96.0	0.588
8.0	4.0	6.0	97.6	0.588
8.0	4.0	7.0	98.6	0.588
8.0	4.0	8.0	99.0	0.588
8.0	6.0	4.0	95.0	0.512
8.0	6.0	5.0	97.1	0.512
8.0	6.0	6.0	98.5	0.512
8.0	6.0	7.0	99.2	0.512
8.0	6.0	8.0	99.4	0.512
8.0	8.0	4.0	96.6	0.493
8.0	8.0	5.0	98.2	0.493
8.0	8.0	6.0	99.1	0.493
8.0	8.0	7.0	99.4	0.493
8.0	8.0	8.0	99.6	0.493

B. RSM

The approach of RSM[8] is to perform a series of experiments, or numerical analyses, for a prescribed set of design points, and to construct a response surface of the measured quantity over the design space. In the present context, the two responses of interest are a measure of combustor performance ERE and the injector wall heat flux Q. The design space consists of the set of relevant variables O/F, V_f/V_0, and L_{comb} considered over the ranges shown in Table 1. The response surfaces are fit by standard least-squares regression with a quadratic polynomial using JMP[9] statistical analysis software. JMP is an interactive, spreadsheet-based program that provides a variety of statistical analysis functions. A backward elimination procedure based on t-statistics is used to discard terms and improve the prediction accuracy. The t-statistic, or t-ratio, of a particular coefficient is given by the value of the coefficient divided by the standard error of the coefficient. The quality of fit between different surfaces can be evaluated by comparing the adjusted rms error defined as

$$\sigma_a = \sqrt{\sum e_i^2/(n-n_p)} \tag{1}$$

where σ_a is the adjusted rms error incurred while mapping the surface over the data set. The measure of error given by σ_a is normalized to account for the degrees of freedom in the model. This rms error, thus, accounts for the nominal effect of higher-order terms providing a better overall comparison among the different surface fits.

In the current study, it is desirable to simultaneously maximize ERE and minimize Q. One method of optimizing multiple responses simultaneously is to build, from the individual responses, a composite response known as the desirability function. The method allows for a designer's own priorities on the response values to be built into the optimization procedure. The first step in the method is to develop desirability function d for each response. In the case where a response should be maximized, such as ERE, the desirability takes the form

$$d_1 = [(ERE - A)/(B - A)]^s \tag{2}$$

where A is the lowest acceptable value such that $d = 1$ for any $ERE > B$ and $d = 0$ for $ERE < A$. The power values s is a weighting factor, which is set according to one's subjective impression about the role of the response in the total desirability of the product. In the case where a response is to be minimized, such as Q, the desirability takes on the form

$$d_2 = [(Q - E)/(C - E)]^t \tag{3}$$

where E is the highest acceptable value such that $d = 1$ for any $Q < C$ and $d = 0$ for $Q > E$. Choices for A, B, C, and E are made according to the designer's priorities or, as in the present study, simply as the boundary values of the domain of ERE and Q spanned by the points in Tables 2–4. Values of s and t are set based on which response has higher priority. A single composite response is developed that is the geometric mean of the desirabilities of the individual responses. The composite response defined as

$$D = (d_1 \cdot d_2 \cdot d_3 \cdots d_m)^{1/m} \tag{4}$$

which for the present case is

$$D = (d_1 \cdot d_2)^{\frac{1}{2}} \tag{5}$$

This is then submitted to an optimization toolbox to be maximized. Solver, an optimization tool available as part of Microsoft Excel[10] package, is used in this effort. This tool uses the GRG2 nonlinear optimization code developed by Lasdon et al.[11]

C. NN

NN[12,13] have received much attention in engineering applications in the last decade because they are hihgly flexible and have the ability to be trained, using user-supplied data, to map complex surfaces. The NN can be trained with data from any source: empirical, experimental, or analytical. Training is accomplished by adjusting weights on the internal connections of the network through defined training algorithms. The training is a cyclic process in which the weights and biases are repeatedly adjusted until an accurate mapping of the response data is obtained. Once trained, the NN is then able to predict the responses for other points in the design space. The NN toolbox available in MATLAB®[14] is used in the current work. A two-layered radial basis network is used to provide the mapping between the input parameters (independent variables) and the output parameters (dependent variables). The network in this effort is designed with the function Solverbe and simulated with Simurb, both of which are contained in the MATLAB NN toolbox. Solverbe designs the network with zero error on the training vectors. It uses the known set of inputs and target vectors along with a quantity called the spread constant to generate the weights and biases for the exact mapping of the network. The designed network has two layers: an initial radial basis layer and a final linear layer. In the initial layer, Solverbe creates as many radial basis neurons as there are input vectors. Each neuron is assigned a weight that is set to the transpose of a given input vector. By design, each neuron detects and responds to a different input vector. Hence, there are as many neurons as input vectors. The radial basis function (*radbas*) has a maximum output of 1 when the input is 0. The radial basis output a is given by

$$a = radbas\{dist(\boldsymbol{w}, \boldsymbol{p}) \times b\} \tag{6}$$

where *radbas* is the transfer function, *dist* is the vector distance between its weight vector $\boldsymbol{w}$ and the input vector $\boldsymbol{p}$, and b is the bias. The bias controls the sensitivity of the neuron. The output has an inverse relationship with the distance between the vectors $\boldsymbol{p}$ and $\boldsymbol{w}$. Any neuron in the network with input identical to its weight vector has an output value of 1. For an input of 0.8326, radial basis produces an output of 0.5. To obtain an output of 0.5 or more, the vector distance between an input vector and its weight vector must be $0.8326/b$ or less. Each bias is set to a value of $0.8326/sc$, where sc is the spread constant. The sc, therefore, defines the range within which the input vector has to lie relative to the weight vector to produce an output of 0.5 or more from the radbas. For large values of sc, neurons should respond strongly to the overlapping regions of the design space. Note that caution must be used regarding selection of values for sc. If the sc value is too large, all neurons may respond to a given input. This creates an erroneous signal, which may adversely affect the network's ability to predict new design points. As discussed in a later section, a study has been conducted to estimate the best value of sc for the present work on injector optimization. Based on the output from the radbas, the second linear layer of the network attempts to map it to the output while minimizing the sum-squared error. Weights and biases are assigned to each neuron based on its output from the radial basis layer such that the network yields a value sufficiently close to the target vector. After the weights and biases have been generated, the MATLAB function Simurb is employed. This function uses the weights and biases generated by Solverbe during the training period to predict the output for new sets of inputs.

The values of independent and dependent variables in Tables 2–4, which constitute the original data set, are used as the inputs and outputs to train the NN. This trained network is then used to generate other design points as required for the enhanced data set.

The design function and design parameters have been found to play an important role in the design of an efficient network. An attempt was made to study the effect of design function. In addition to Solverbe, another MATLAB design function, Solverb, was used to repeat the training/prediction procedure described. A significant difference was noticed in the prediction capabilities of the two functions. As compared to Solverbe, Solverb adds one neuron at a time instead of adding as many neurons as the number of inputs. It is an iterative procedure, and neurons are added until the error during training is less than the user-defined error goal. This iterative procedure of adding neurons one at a time may result in a smaller network, but it takes a longer time to train than Solverbe. It also requires more design parameters than Solverbe.

More detailed discussion of the basic concepts and practical implementation of both RSM and NN, directly relevant to the present effort, can be found in Ref. 13.

IV. Results and Discussion

A. Polynomial Fits on the Original Data Set

According to the injector model developed by Calhoon et al.,[7] injector performance, as measured by *ERE* depends only on the velocity ratio V_f/V_0 and combustion chamber length L_{comb}. Examination of the original data set in Tables 2–4 indicates 15 distinct design points for *ERE*. Because chamber wall heat flux is dependent only on velocity ratio V_f/V_0 and oxidizer to fuel ratio O/F, there are nine distinct design points for Q. The design space for this effort is shown in Fig. 1. For *ERE*, the five distinct chamber lengths offer the potential for a fourth-order polynomial fit in L_{comb}, whereas the three different velocity ratios limit the fit in V_f/V_0 to second order. Quadratic and cubic response surfaces for both *ERE* and Q has been generated for evaluation. These noted limitations on the data cause the cubic surfaces to be third order in L_{comb} only.

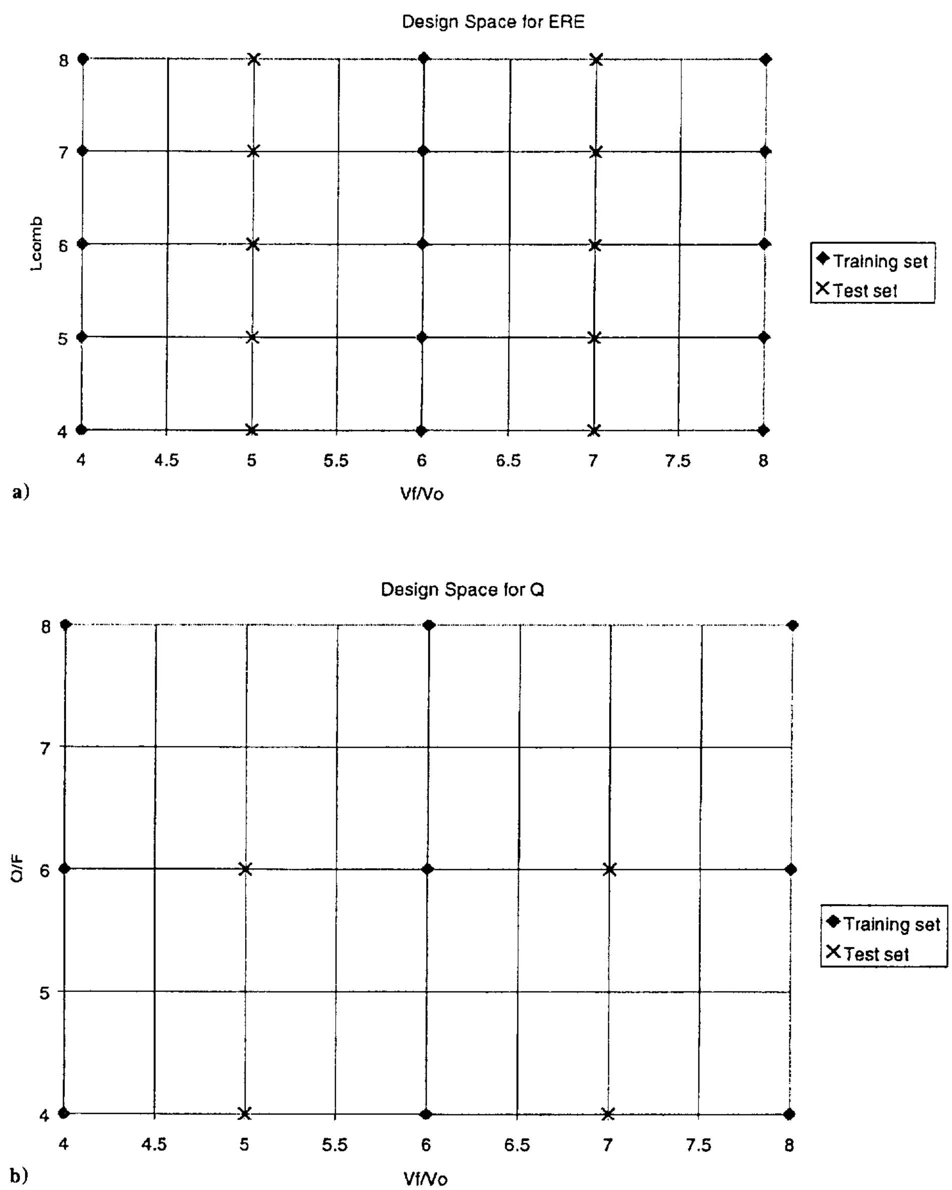


Fig. 1 Design space for a) *ERE*, 15 training/mapping points, and 10 test points and b) Q, 9 training/mapping points and 4 test points.

1. Quadratic Response Surface

The quadratic response surfaces on the original data set constructed by JMP are given by

$$ERE = 70.43 + 1.580 V_f/V_0 + 6.208 L_{\text{comb}} - 0.190(V_f/V_0)L_{\text{comb}} - 0.331(L_{\text{comb}})^2 \tag{7}$$

and

$$Q = 0.479 - 0.046 O/F + 0.191 V_f/V_0 + 0.0094(O/F)^2 - 0.028(O/F)V_f/V_0 \tag{8}$$

These response surfaces represent reduced models accomplished by term elimination from the full surface using t statistics as described earlier. These are identical to reduced surfaces generated previously by Tucker et al.[6]

2. Cubic Response Surface

Cubic models for *ERE* and Q have also been generated for the analysis. The full *ERE* and Q response surfaces are

$$ERE = 50.060 + 3.759 V_f/V_0 + 14.574 L_{\text{comb}} - 0.05(V_f/V_0)^2 - 0.777(V_f/V_0)L_{\text{comb}} - 1.459(L_{\text{comb}})^2 + 0.0025(V_f/V_0)^2 L_{\text{comb}} + 0.0464 V_f/V_0(L_{\text{comb}})^2 + 0.0472(L_{\text{comb}})^3 \tag{9}$$

and

$$Q = -0.566 - 0.358 O/F + 0.383 V_f/V_0 - 0.0191(O/F)^2 - 0.107(O/F)V_f/V_0 - 0.0028(V_f/V_0)^2 + 0.0048(O/F)^2 V_f/V_0 + 0.0019(O/F)(V_f/V_0)^2 \tag{10}$$

Reduced cubic surfaces for both *ERE* and Q were also obtained. As discussed later, shortcomings were encountered with these surfaces.

3. Comparison Between Cubic and Quadratic Response Surfaces

The quadratic and cubic fits for both surfaces are plotted along with the actual data from the injector model in Fig. 2. NN and injector model data are the same points in the graph. Quadratic and cubic are predicted by RSM. Based on the adjusted rms error, the cubic fit is more accurate than the quadratic fit for *ERE*. The adjusted rms error for the quadratic and cubic response surfaces of *ERE* are 0.211 and 0.083, respectively. The cubic fit, by this measure, is superior for *ERE*. However, the error is almost identical in the case of Q for both the quadratic (0.039) and cubic (0.040) surfaces. This maybe due to the very small number of points available for the curve fit. The additional terms in the cubic fit relative to the quadratic fit do not improve the mapping of the response surface for Q.

In the report by Tucker et al.,[6] an optimization was done for three different ranges of the independent variables using the quadratic fit shown in Eqs. (7) and (8). The three cases analyzed differ only in the constraints implemented on the design parameters. The

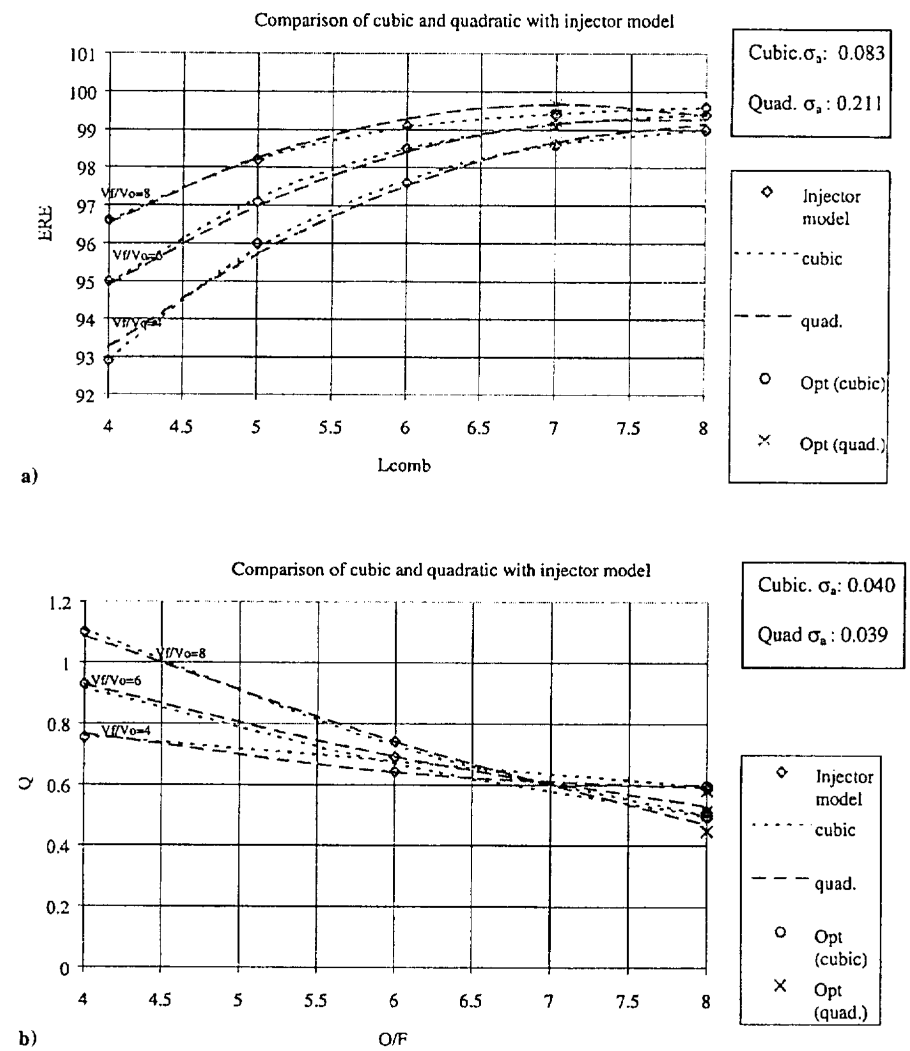


Fig. 2 Assessment of performance of cubic and quadratic response surfaces of a) *ERE*, 15 training/mapping points and b) Q, 9 training/mapping points.

Table 5 Optimum values obtained with cubic and quadratic for case 1[a]

		Cubic					Quadratic				
W_{ERE}, (s)	W_Q, (t)	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s
1	10	6.0	**5.41**	7.0	99.02 (99.00)[b]	0.664 (0.654)	6.0	**6.00**	7.0	99.17 (99.20)	0.669 (0.642)
1	1	6.0	6.00	7.0	99.15	0.669	6.0	6.00	7.0	99.17	0.669
10	1	6.0	6.00	7.0	99.15	0.669	6.0	6.00	7.0	99.17	0.669

[a]Constraints: $4 \leq O/F \leq 6$, $4 \leq V_f/V_0 \leq 6$, and L_{comb} (in.) ≤ 7. [b](Exact response of the injector model.)

Table 6 Optimum values obtained with cubic and quadratic for case 2[a]

		Cubic					Quadratic				
W_{ERE}, (s)	W_Q, (t)	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s
1	10	6.0	**5.41**	7.0	99.02 (99.00)[b]	0.664 (0.654)	6.0	**6.52**	7.0	99.31 (99.10)	0.684 (0.716)
1	1	6.0	**6.34**	7.0	99.21 (99.20)	0.674 (0.691)	6.0	**7.00**	7.0	99.42 (99.30)	0.702 (0.728)
10	1	6.0	7.00	7.0	99.32	0.690	6.0	7.00	7.0	99.42	0.702

[a]Constraints: $4 \leq O/F \leq 6$, $5 \leq V_f/V_0 \leq 7$, and L_{comb} (in.) ≤ 7. [b](Exact response of the injector model.)

Table 7 Optimum values obtained with cubic and quadratic for case 3[a]

		Cubic					Quadratic				
W_{ERE}, (s)	W_Q, (t)	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s
1	10	6.0	**6.00**	7.0	99.15	0.669	6.0	**6.52**	7.0	99.31	0.684
1	1	6.0	**6.34**	7.0	99.21	0.674	6.0	**8.00**	7.0	99.67	0.753
10	1	6.0	8.00	7.0	99.42	0.728	6.0	8.00	7.0	99.67	0.753

[a]Constraints: $4 \leq O/F \leq 6$, $6 \leq V_f/V_0 \leq 8$, and L_{comb} (in.) ≤ 7.

constraints are as follows: case 1, $4 \leq O/F \leq 6$, $4 \leq V_f/V_0 \leq 6$, and $L_{comb} \leq 7$; case 2, $4 \leq O/F \leq 6$, $5 \leq V_f/V_0 \leq 7$, and $L_{comb} \leq 7$; and case 3, $4 \leq O/F \leq 6$, $6 \leq V_f/V_0 \leq 8$, and $L_{comb} \leq 7$.

In the current effort, the optimization is repeated using the cubic fits in Eqs. (9) and (10). The combinations of weights for ERE (s) and Q (t) used are (1,10), (1,1), and (10,1) for each of the three cases. The optimum has been evaluated and tabulated for each case for each of the three weightings. Tables 5–7 show the results for the 18 resulting optimization exercises. Recall, that in this effort, injector element optimization means maximizing the performance while minimizing heat flux and chamber length. The optimum value for V_f/V_0 obtained on the cubic response surface is quite different than that found on the quadratic surface for some cases (these particular cases are noted in bold face in Tables 5–7). Also, for selected cases where there are discrepancies between the quadratic and cubic results, the exact values from the injector model have been included in parentheses in Tables 5–7 for comparison. In these cases, the cubic fit more closely matches the exact data than does the quadratic fit. Sample results for *ERE* plotted in Fig. 3a clearly show the data are better fit by the cubic surface for the case shown. Figure 3b shows that the response surface predicted by cubic fit for Q has a noticeable dip that is completely missed by the quadratic fit. This discrepancy results in the optimum for the cubic fit being considerably lower than that for the quadratic surface. The prediction from cubic fit agrees well with the exact data, which also has a dip for this specific case. NN and injector model data are the same points in Fig. 3. Quadratic and cubic are predicted by RSM.

The injector model was also used to produce additional design points to assess the capability of the different response surfaces to match the exact data. In Figs. 4a and 5a, the actual data obtained from the injector model for all of the design points are shown. The cubic and quadratic response surfaces obtained based on the original data is also shown. The rms error σ for the test data is given by

$$\sigma = \sqrt{e_i^2/n} \tag{11}$$

In this equation, n is the number of test points. The rms error for predicting the new *ERE* data is 0.270 and 0.205 for the quadratic and cubic surfaces, respectively. For Q, it is 0.025 and 0.016 for the quadratic and cubic surfaces, respectively. Again, the performance of the cubic surface is superior to that of the quadratic surface.

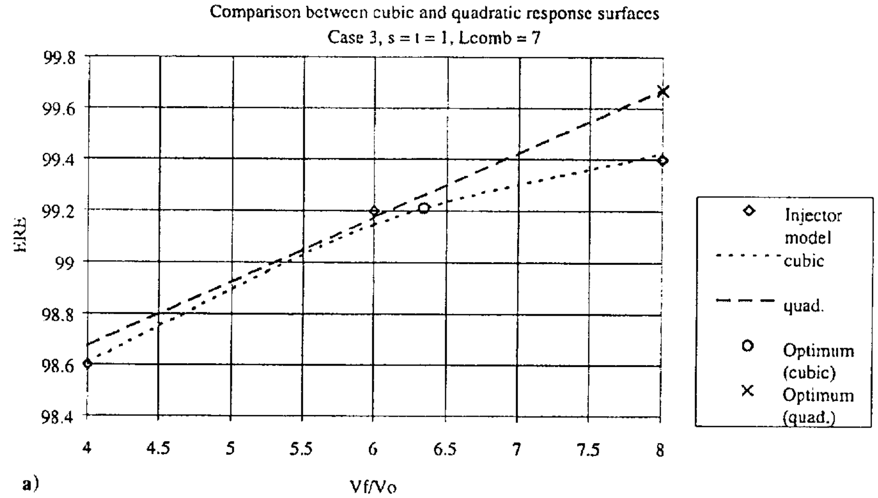
A reduced cubic model was also obtained, but the difference in the adjusted rms error was small as compared to the full model. It was found to be 0.077 for *ERE* and 0.042 for Q as compared to the value of 0.083 for *ERE* and 0.040 for Q for the full model. Despite being comparable to the full model from the standpoint of the adjusted rms error, it was found that the rms error in predicting the new data was significantly degraded for Q. The rms error in predicting the new data was 0.203 for *ERE* and 0.038 for Q for the reduced models as compared to 0.205 for *ERE* and 0.016 for Q for the full cubic model. Accordingly, the full model was preferred to the reduced one.

B. RBNN

RBNN are trained by both Solverbe and Solverb for each injector design response, *ERE* and Q, using the original data set of 45 design points. Solverbe trained the network for *ERE* with an error to the order of 10^{-13}. The network trained by Solverbe for Q has an error on the order of 10^{-16}. Both networks represent the respective design spaces essentially exactly. Solverb, with an error goal of 0.001, trained networks for both responses to represent the original data set adequately. Because the size of the data set considered for training the network is fairly modest, the number of neurons generated by Solverbe is also small. Solverb would have been suited better for a larger data set, where a reduction in the number of neurons might have appreciably reduced the computation time. The networks trained using Solverbe have been used for this study. The ability of the RBNN to fit the design data and generate additional data is discussed in the following sections.

1. Comparison Between Solverbe and Solverb

Because Solverbe trains with the same number of neurons (45 in this case) as data points, as seen earlier, it fits the training data set



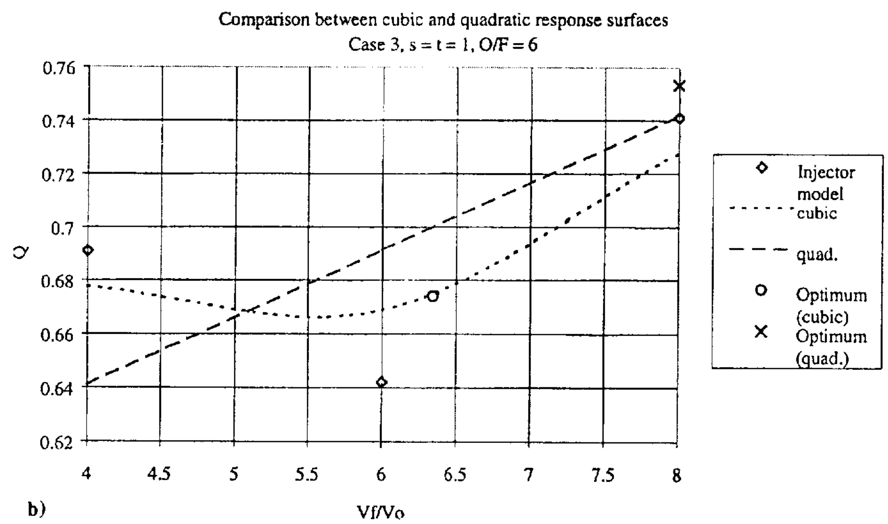


Fig. 3 Comparison between cubic and quadratic response surface for case 3 of a) *ERE* and b) *Q*.

with negligible error. However, it can also create erratic behavior because it makes no attempt to filter noise generated by excess neurons in the network. Solverb, on the other hand, tends to reduce the potential for noise by controlling the number of neurons in the network. Table 8 shows that in the present study, for the spread constant value of 1.00, Solverb performs slightly better than Solverbe based on the nominal error measure. However, when judged by the level of errors associated, both RBNNs are satisfactory from a practical standpoint. As expected, Solverb uses fewer neurons than Solverbe, in this case three less. Note that, as investigated in detail by Papila et al.,[15] the relative performance between Solverb and Solverbe is case dependent.

Table 8 RMS error in the prediction of ERE and *Q*. Different values of SC

	Solverbe		Solverb[a]		
sc	rms error (*ERE*, %)	rms error (Q, Btu/in.^{2}-s)	rms error (*ERE*,%)	rms error (Q, Btu/m.^{2}-s)	No. of neurons
0.50	1.493	0.179	1.733	0.287	44
0.75	0.745	0.135	0.675	0.135	44
1.00	**0.152**	**0.022**	**0.153**	**0.017**	**42**
1.05	**0.190**	**0.011**	**0.128**	**0.012**	**44**
1.25	0.316	0.010	0.267	0.022	44
1.50	0.336	0.022	0.309	0.030	44
1.75	0.369	0.022	0.310	0.021	44
2.00	0.308	0.016	0.296	0.019	41
2.25	0.279	0.020	1.846	0.045	43
2.50	0.325	0.017	0.744	0.025	43

[a]Error goal used for Solverb is 0.001.

2. *Comparison of RBNN Predictions with RSM*

Figures 4b and 5b show that the RBNN trained by Solverbe is able to generate more accurately additional design data than either quadratic or cubic polynomial (shown for comparison in Figs. 4a and 5a). In Fig. 4a, the *ERE* surface trained with the original data set is shown. The 10 extra design points calculated with the injector model for V_f/V_0 of 5.00 and 7.00 are shown. The ability of the RBNN to generate accurately new design data can be seen by comparing the fit for *ERE* in Fig. 4b to that for the polynomials in Fig. 4a. RBNN trains the network with more flexibility and learns the data trend, whereas polynomials provide only an approximate fit on the given data. Regarding the rms error σ, for *ERE*, it is 0.152 for RBNN predictions as compared to the values of 0.270 and 0.205 for quadratic and cubic surfaces, respectively. The four extra design points generated for Q, also at V_f/V_0 of 5.00 and 7.00, are shown compared to the polynomial surface in Fig. 4b and compared to the RBNN surface in Fig. 5b. The rms error in the case of Q is 0.022 for RBNN as compared to 0.025 and 0.016 for quadratic and cubic surfaces, respectively. Here the performance of the RBNN is better than the quadratic but slightly poorer than the cubic fit. Examination of Table 8 indicates it may be possible that using Solverb with a spread constant of 1.05 could further reduce the rms for Q. However, the errors for Q are low enough that further reduction may not be practical.

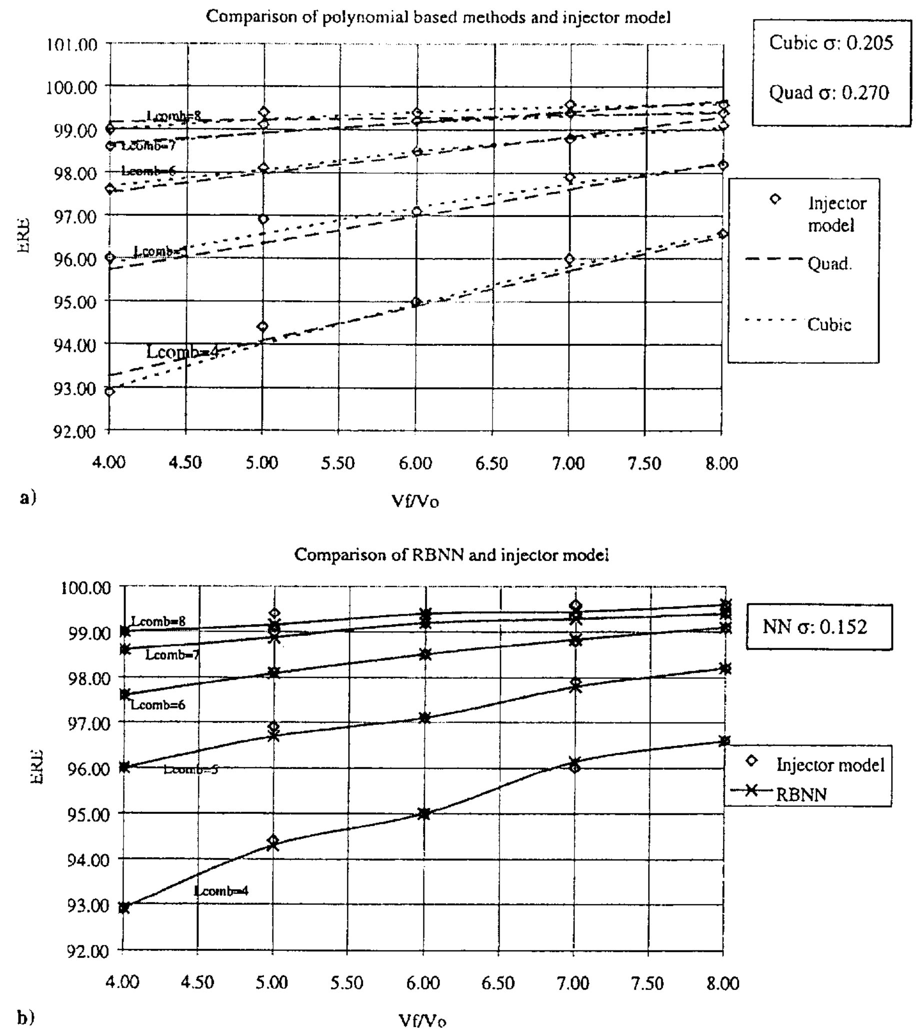


Fig. 4 Assessment of predictive capabilities of a) polynomial based method and b) RBNN for *ERE*, 15 training/mapping points 10 test points.

C. RBNN-Enhanced Response Surface

It has been demonstrated that the RBNN can be used to generate confidently additional design points. Additional design points generated by the RBNN are added to the original data set to form the enhanced data set. This enhanced data set is used for further analysis to evaluate the performance of the RSM with the larger number of design points. The enhanced data set for *ERE* has 15 points from the injector model and 10 from the RBNN, for a total of 25 points. The enhanced data set for Q has 9 points from the injector model and 4 from the RBNN, for a total of 13 points. The entire optimization analysis was redone with the enhanced data set. On this enhanced data set, the full quadratic response surface seems already appropriately constructed, and invoking the statistical analysis generates no reduced model. With the added data in the enhanced data set, it is now possible to obtain a fit for *ERE* that is fourth order in V_f/V_0 and fourth order in L_{comb}. Q can now be fit with a cubic in V_f/V_0 and a quadratic in O/F. This is now possible because a combination of three different values of O/F, five different values of V_f/V_0, and five different values of L_{comb} are available. The cubic fit for *ERE* and Q obtained from JMP are

$$\begin{aligned} ERE = {} & 48.813 + 4.807 V_f/V_0 + 14.274 L_{\text{comb}} - 0.141(V_f/V_0)^2 \\ & - 0.930(V_f/V_0)L_{\text{comb}} - 1.352(L_{\text{comb}})^2 \\ & + 0.0003(V_f/V_0)^3 + 0.0154(V_f/V_0)^2 L_{\text{comb}} \\ & + 0.0463 V_f/V_0(L_{\text{comb}})^2 + 0.042(L_{\text{comb}})^3 \end{aligned} \tag{12}$$

and

$$\begin{aligned} Q = {} & -0.301 + 0.323 O/F + 0.285 V_f/V_0 - 0.0189(O/F)^2 \\ & - 0.094(O/F)V_f/V_0 + 0.00475(V_f/V_0)^2 \\ & + 0.00474(O/F)^2 V_f/V_0 + 0.0008(O/F)(V_f/V_0)^2 \\ & + 0.000058(V_f/V_0)^3 \end{aligned} \tag{13}$$

The quadratic fits from JMP are

$$\begin{aligned} ERE = {} & 69.68 + 2.088 V_f/V_0 + 6.024 L_{\text{comb}} - 0.042(V_f/V_0)^2 \\ & - 0.190(V_f/V_0)L_{\text{comb}} - 0.139(L_{\text{comb}})^2 \end{aligned} \tag{14}$$

and

$$\begin{aligned} Q = {} & 0.812 - 0.045 O/F + 0.067 V_f/V_0 + 0.0097(O/F)^2 \\ & - 0.028(O/F)V_f/V_0 + 0.0105(V_f/V_0)^2 \end{aligned} \tag{15}$$

1. *Comparison of Fits with the Original Response Surfaces*

Comparison of the enhanced response surfaces with the original response surfaces indicates that the extra data produced with the RBNN generally improve the quality of the curve fit. The adjusted rms error for *ERE* on the original set is 0.211 and 0.083 for quadratic and cubic fits, respectively. On the enhanced data set, it is 0.179 and 0.100 for the quadratic and cubic fits, respectively. The slight increase in the error in the case of the cubic fit may be due to

Table 9 Optimum values obtained with cubic and quadratic for case 1 (enhanced data set)[a]

		Cubic					Quadratic				
W_{ERE}, (s)	W_Q, (t)	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	ERE, %	Q, Btu/in.^{2}-s
1	10	6.0	**5.54**	7.0	99.02 (98.90)	0.654 (0.658)	6.0	**5.01**	7.0	98.96 (98.70)	0.644 (0.664)
1	1	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658
10	1	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658

[a]Constraints: $4 \leq O/F \leq 6$, $4 \leq V_f/V_0 \leq 6$, and L_{comb} (in.) ≤ 7. Compare with Table 5.

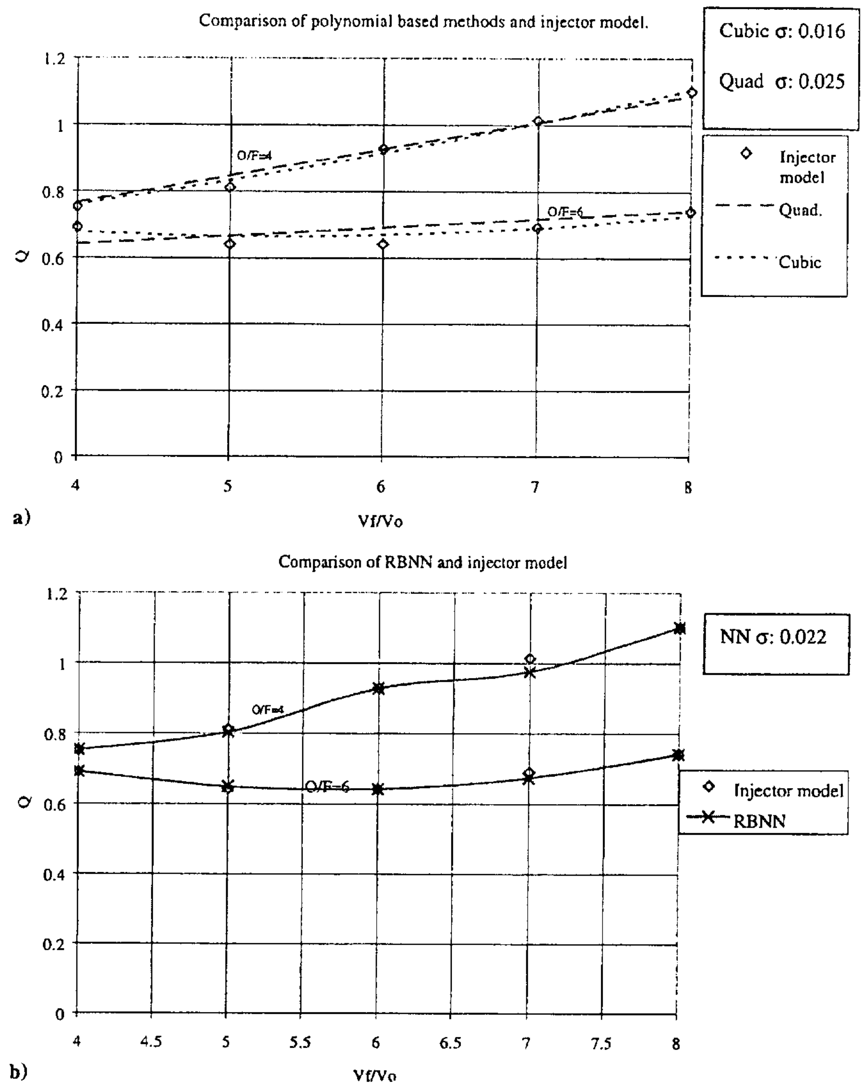


Fig. 5 Assessment of predictive capabilities of a) polynomial based methods and b) RBNN for Q, 9 training/mapping points, 4 test points.

noise related to the oversensitivity of the polynomial. However, this phenomenon may reflect that the level of the rms is low enough in either case so that no further improvement is accomplished. The adjusted rms error for Q with the original set is 0.039 and 0.040 for the quadratic and cubic fits, respectively. On the enhanced set it was 0.027 and 0.026 for the quadratic and cubic, respectively. With the exception of the cubic fit for *ERE*, the fits from the enhanced surface are improved over those from the original surface. Also, when optimum design points are examined, there is less difference between the quadratic and cubic fits on the enhanced surfaces than there is on the original surfaces.

2. *Comparison of Optimal Design Points*

The analysis for the three cases of optimization over the same three ranges of independent variables has been redone. The results of the optimization on surfaces generated from the enhanced data set are given in Tables 9–11. The predicted optimal design points using cubic and quadratic fits are generally close to each other. They are closer to each other on the reduced data set than on the surfaces generated using the original data set. One case where the cubic and quadratic optimum points are somewhat different is analyzed further. The results shown in Fig. 6 confirm the optimum value of velocity ratio on the quadratic fit to be lower than the cubic fit in this case. The enhanced set includes model data and RBNN predicted data. Quadratic and cubic are predicted by RSM. Given the weightings of 1.0 for *ERE* and 10.0 for Q, the optimizer has selects the minimum of Q for both fits. Because the curves exhibit different minimum points, the weightings force the selection of different optimum points. As already discussed, for the polynomial fits on the RBNN-enhanced data sets, the error of both quadratic and cubic

Table 10 Optimum values obtained with cubic and quadratic for case 2 (enhanced data set)[a]

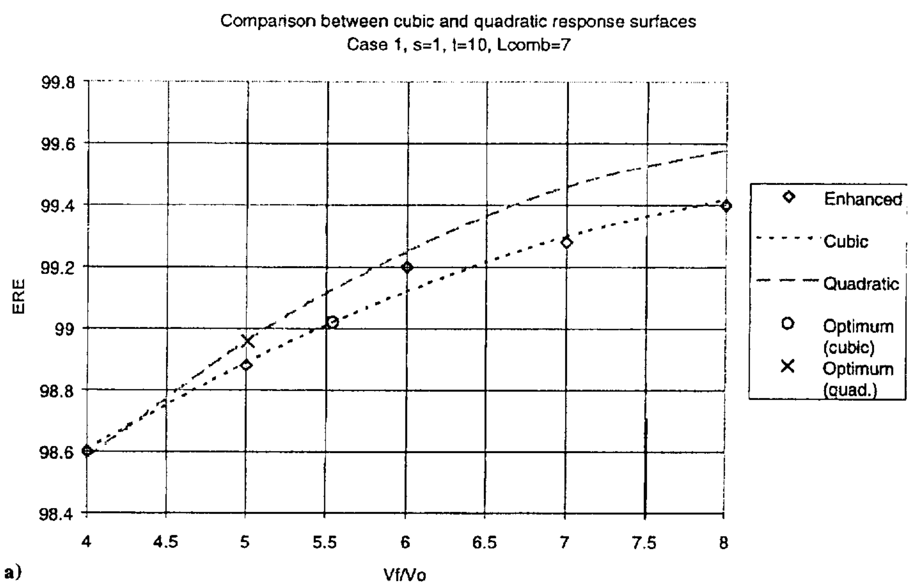
		Cubic					Quadratic				
W_{ERE}, (*s*)	W_Q, (*t*)	O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	*Q*, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	*Q*, Btu/in.^{2}-s
1	10	6.0	**5.54**	7.0	99.02 (98.90)	0.654 (0.658)	6.0	**5.01**	7.0	98.96 (98.70)	0.644 (0.664)
1	1	6.0	**6.33**	7.0	99.18 (99.10)	0.663 (0.666)	6.0	**6.04**	7.0	99.26 (99.20)	0.659 (0.642)
10	1	6.0	7.00	7.0	99.30	0.681	6.0	7.00	7.0	99.46	0.693

[a]Constraints: $4 \leq O/F \leq 6, 5 \leq V_f/V_0 \leq 7$, and L_{comb} (in.) ≤ 7. Compare with Table 6.

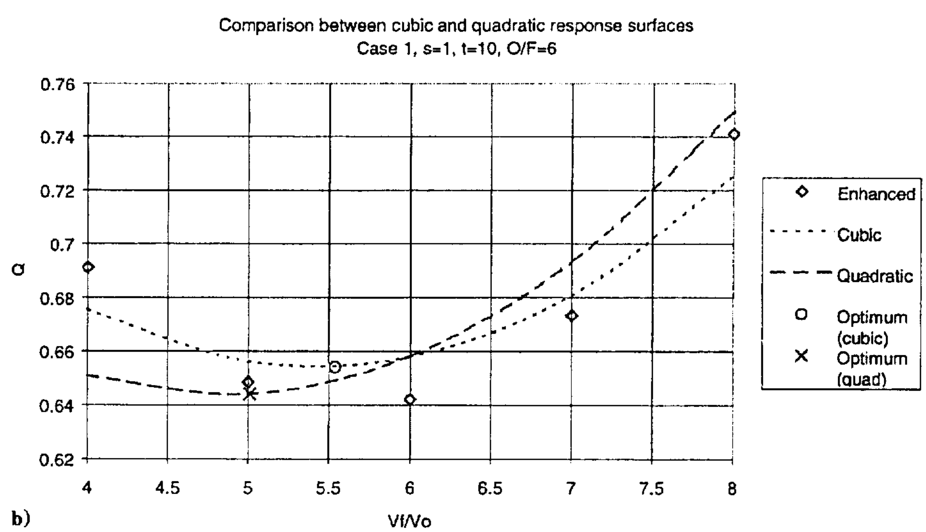
Table 11 Optimum values obtained with cubic and quadratic for case 3 (enhanced data set)[a]

		Cubic					Quadratic				
W_{ERE}, (*s*)	W_Q, (*t*)	O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	*Q*, Btu/in.^{2}-s	O/F	V_f/V_0	L_{comb}, in.	*ERE*, %	*Q*, Btu/in.^{2}-s
1	10	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658
1	1	6.0	**6.33**	7.0	99.19	0.663	6.0	**6.04**	7.0	99.26	0.659
10	1	6.0	8.00	7.0	99.42	0.725	6.0	7.95	7.0	99.57	0.746

[a]Constraints: $4 \leq O/F \leq 6, 6 \leq V_f/V_0 \leq 8$, and L_{comb} (in.) ≤ 7. Compare with Table 7.



a)



b)

Fig. 6 Assessment of performance of cubic and quadratic response surfaces for case 1 of a) *ERE*, 25 training/mapping points and b) *Q*, 13 training/mapping points.

polynomials are more comparable than in the original analysis. At the upper limit of available data for combustor length, 8 in., the *ERE* curves tend to flatten out. This causes some difficulty in locating the optimum and may cause more noticeable differences between the different polynomials. However, different optimal designs selected by different polynomials under such a circumstance are not important because these yield very similar injector performance.

V. Summary and Conclusions

RSM and NN techniques have been applied to the optimization of a simple rocket injector. Injector performance, as measured by *ERE* and chamber wall heat flux, has been modeled as a function of propellant velocity ratio, oxidizer to fuel ratio, and combustor length. An empirical injector model was used to calculate 45 design points in the design space. The responses in these original data were fit to the input variables using both quadratic and cubic polynomials using RSM as embodied in the JMP software package. The fits were evaluated relative to each other using the adjusted rms error and the ability of the fit to predict additional data from the empirical model. Optimization studies were conducted using each fit over three ranges of independent variables for different weightings, or desirability, of the responses. Optimum points were located for each of 18 combinations of data range, curve fit, and response weighting using Microsoft Excel-SOLVER. These optimum points were then compared to exact values calculated from the empirical model.

RBNN were trained on the original data set using functions from the MATLAB-NN toolbox. Issues relevant to obtaining satisfactory RBNN performance and to enhancing the RSM for the current problem were investigated. An RBNN was trained on the original data set. The RBNN was compared to the RSM in terms of ability to predict additional data in the design space. The RBNN was then used to generate new data, which were combined with the original data set to form an enhanced data set. The optimization procedure was repeated using the enhanced data set. Quality of fit and location of optimal points were used to compare the fits from the enhanced surface with those from the original data set.

Based on the effort described, the following observations are made for the present injector design system.

1) The cubic fit was superior to the quadratic fit on the modest-sized original data set by each measure investigated: first, in terms of adjusted rms error, second, in that the optimal design points were closer to the data from the empirical model, and finally, in terms of the rms error relative to predicting additional data in the design space.

2) There was not a significant difference in the performance of Solverbe and Solverb in terms of generating the RBNN on the original data set.

3) The RBNN was able to generate additional design data for ERE with better accuracy than either the quadratic or cubic fit. For Q, the RSM error from both polynomials on the original data set and for the RBNN were all very low. The rms error for the RBNN fell between that of the two polynomials.

4) Comparison of the quality of polynomial fits on the original and RBNN-enhanced data sets indicated generally better fits on the enhanced surface. The only exception was that the cubic fit for *ERE* was slightly poorer on the enhanced surface. However, the error was already small.

5) The optimal points located on the quadratic and cubic surfaces generated from the enhanced data set were, for the same case, consistently closer to each other than were the fits from the original data set. Also, the original data set cubic fit was closer to the enhanced data set cubic fit than were the quadratic fits for the two data sets.

The preceding observations, taken together, indicate that RSM, when used in conjunction with NN, is capable of producing meaningful optimization studies with modest amounts of data. NN can be used to produce data of sufficient accuracy to actually improve the quality of polynomial fit in the RSM. Because accurate data (either from physical tests or CFD analysis) are time consuming and expensive to obtain for rocket engine injectors, the technique of coupling RSM with NN holds significant potential for their optimization.

Acknowledgment

The present work has been supported in part by NASA Marshall Space Flight Center.

References

[1]Rupe, J. H., "An Experimental Correlation of the Nonreactive Properties of Injection Schemes and Combustion Effects in a Liquid Rocket Engine," NASA TR 32-255, 1965.

[2]Pieper, J. L., "Oxygen/Hydrocarbon Injector Characterization," U.S. Air Force Phillips Lab., PL-TR 91-3029, Oct. 1991.

[3]Nurick, J. H., "DROPMIX-A PC Based Program for Rocket Engine Injector Design," *JANNAF Propulsion Conference*, WY, Nov. 1990.

[4]Dickerson, R., Tate, K., and Nurick, W., "Correlation of Spray Injector Parameters with Rocket Engine Performance," U.S. Air Force Rocket Propulsion Lab., AFRPL-TR-68-11, Jan. 1968.

[5]Pavli, A. L., "Design and Evaluation of High Performance Rocket Engine Injectors for use with Hydrocarbon Fuels," NASA TM 79319, 1979.

[6]Tucker, P. K., Shyy, W., and Sloan, J. G., "An Integrated Design/Optimization Methodology for Rocket Engine Injectors," AIAA Paper 98-3513, July 1998.

[7]Calhoon, D., Ito, J., and Kors, D., "Investigation of Gaseous Propellant Combustion and Associated Injector-Chamber Design Guidelines," Aerojet Liquid Rocket Co., NASA CR-121234, Contract NAS3-13379, July 1973.

[8]Myers, R. H., and Montgomery, D. C., *Response Surface Methodology—Process and Product Optimization Using Designed Experiment*, Wiley New York, 1995, pp. 16–67, 208–265.

[9]JMP, Ver. 3, SAS Inst., Inc., Cary, NC, 1995.

[10]Microsoft Excel 97, Microsoft Corp., WA, Copyright 1985–1996.

[11]Lasdon, L. S., Waren, A., Jain, A., and Ratner, M., "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 4, No. 1, 1978, pp. 34–50.

[12]Hertz, J., Krogh, A., and Palmer, R. G., *Introduction to the Theory of Neural Computation*, Lecture Notes, Sante Fe Inst. Studies in the Sciences of Complexity, Addison Wesley Longman, Reading, MA, 1991, pp. 89–156.

[13]Vaidyanathan, R., Papila, N., Shyy, W., Tucker, P. K., Griffin, L. W., Haftka, R. T., and Fitz-Coy, N., "Neural Network and Response Surface Methodology for Rocket Engine Component Optimization," AIAA Paper 2000-4880, Sept. 2000.

[14]Demuth, H., and Beale M., MATLAB Neural Network Toolbox, Mathworks Natick, MA, 1992.

[15]Papila, N., Shyy, W., Fitz-Coy, N., and Haftka, R. T., "Assessment of Neural Net and Polynomial-Based Techniques for Aerodynamic Applications," AIAA Paper 99-3167, June-July 1999.

SENSITIVITY EVALUATION OF PHYSICAL MODELS USING RESPONSE SURFACE METHODOLOGY

Rajkumar Vaidyanathan, Inanc Senocak and Wei Shyy
Department of Aerospace Engineering, Mechanics & Engineering Science
University of Florida, Gainesville, FL 32611-6250 USA

ABSTRACT

A sensitivity analysis is done for turbulent cavitating flows with the aid of response surface methodology. A pressure-based Navier-Stokes solver using a finite volume approach is coupled with a volume transport equation to obtain a cavitation model. The parameters influencing the turbulence and cavitating models are identified. Firstly, a non-cavitating flow over a backward-facing step is analyzed. A non-equilibrium model of the k-ε model is used. The sensitivity of the CFD solutions to the parameters regulating the production and dissipation of the turbulent kinetic energy is evaluated. Secondly, using the information from this study, cavitating flows around a hemispherical and a blunt fore-body are analyzed. Sensitivities of the additional tuning parameters arising from the source terms in the transport equation are also evaluated. In this study, design of experiments is used to select the design space for these parameters. A response surface is created based on the CFD solutions and a search for the region with best performance is carried out. Comparison with experimental results has been done to judge the performance of the CFD solutions. This study intends to come up with a general scheme, which can aid in the selection of the tuning parameters for identical cavitating flows.

INTRODUCTION

The inception of cavitation occurs due to the drop in pressure of the flow below the vapor pressure of the liquid[1, 2, 3, 4]. This kind of situation is noticed in liquid flows, which would involve rotating flows like in pumps, nozzles, etc, and around underwater bodies. These flows are highly turbulent and involve the physics of multiphase flow. Cavitation results in structural damage to the body on which it occurs, along with a decrease in their performance. On the other hand it has also been an interesting topic of research due to the reduction in drag on the body when it is completely covered with a cavity, a flow termed as supercavitating flow.

Recent years have seen a significant progress in computational modeling of cavitation. Reynolds averaged Navier-Stokes equations have been applied along with physical models to represent the cavitation process. The existing studies have shown that modeling of cavitation is a complicated task, as one has to address the physical modeling, the numerical algorithm and turbulence modeling adequately such that any of these three subtasks do not mask or interfere the performance of others. In the context of physical modeling a transport equation model have been proposed[5, 6]. Volume or mass fraction transport equation with source terms is solved to generate the cavitating regions. Basically the source terms regulate the rate of evaporation and condensation. Different studies have proposed different source terms along with empirical constants used to adjust the rate of the processes[5, 6, 7]. The choice of these empirical constants is mainly guided by physical intuition and may depend on the geometry considered[8, 9]. It has also been shown that once a proper combination of the empirical constants is found, increasing these constants by an order of magnitude has little affect on the pressure distribution whereas the density distribution experiences a significant change[8].

Numerical algorithms have also been improved and developed for turbulent cavitating flows. The density-based technique have been applied by several researchers[6, 7, 9, 10]. In these studies the artificial compressibility method is used and a special attention has been given to the preconditioning technique to handle the multiphase nature of cavitating flows. Recently Senocak and Shyy[8] have proposed a pressure-based method for turbulent cavitating flows along the lines of well-established SIMPLE method[11]. In this study, it has been demonstrated that the pressure correction equation exhibits a convective-diffusive character in regions of cavitation. So far both pressure-based and density-based techniques have been successful in matching the experimental data for the cases studied.

Since cavitation is generally considered as a high Reynolds number phenomenon, turbulence modeling plays a key role and must also be addressed adequately in addition to the physical modeling and numerical algorithm. Most of the recent research has focused on physical and algorithmic issues and a standard k-ε model have been used for turbulence closure in these studies. Senocak and Shyy[8] have investigated the effect of turbulence modeling in the context of nonequilibrium k-ε model[12]. In this study, it has been demonstrated that for cavitating flows with large streamline curvature and recirculation zones the turbulence model can influence the performance of cavitation modeling and the issue needs an in depth study.

The original k-ε turbulence model, which is based on equilibrium assumptions, needs to be modified for a flow with rotation, recirculation and large streamline curvatures. The modeling parameters, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$, which control the production and dissipation of the turbulence kinetic energy, respectively, has to be properly regulated. A volume fraction transport equation model is included to account for the cavitation dynamics. In this model there exists two empirical parameters, namely C_{dest} and C_{prod}, to regulate the phase change process. The sensitivities of these four parameters are evaluated with the aid of design of experiments and Response Surface Methodology[13].

To facilitate the evaluation of a turbulent cavitation model, we have adopted a two-stage process. Firstly we evaluate the performance of the nonequilibrium k-ε turbulence model under non-cavitating conditions with the aid of a flow over a backward-facing step. The goal is to identify the interdependency of the key parameters, namely, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$, so as to narrow down the range of variations of these parameters.

Secondly, cavitating flow analysis is done for axisymmetric cylindrical geometries. Two geometries with a hemispherical and a blunt fore-body are used with different cavitation numbers. Cavitation number[1, 2] is used to judge the likelihood of the inception of cavitation. It is defined as

$$\sigma = \frac{2\left(P_{\infty} - P\right)}{\rho U_{\infty}^{2}} \tag{1}$$

Chances of cavitation are more for lower values of σ. This is a measure which gives an idea of the nature of the flow for a given reference pressure or when the flow velocity is changed. For the same cavitation number a considerable difference can be noticed in the cavitating characteristics for different geometries.

A Response Surface Methodology (RSM) is used to evaluate the sensitivity and to identify the optimized model parameters, based on the selected flow conditions and geometries. The RSM approach is to conduct a series of well-chosen experiments (empirical, numerical, physical or some combination of the three) and use the resulting information to construct a global approximation (response surface) of the measured quantity (response) over the design space. A standard constrained optimization algorithm is then used to interrogate the response surface for an optimum design.

Our goals in this study are:

i) Assess the sensitivity of the selected turbulent cavitating flow model to the modeling parameters.
ii) Probe the impact of the individual and collective behaviors of the modeling parameters. This would help in gaining an insight into the range of predictions that we can expect with these models.
iii) Propose a combination for the set of parameters that gives optimum predictive capabilities.

In the following sections, the governing equations used in the study are presented. The turbulence modeling parameters are studied separately by computing a non-cavitating flow over a backward-facing step. Following this, cavitating flow analysis is done for axisymmetric cylindrical geometries. The insight obtained on the turbulence modeling parameters is used to reduce the design space for this study. The inclusion of the transport equation for cavitation introduces additional parameters, which are addressed with the aid of the flow over the mentioned geometries.

GOVERNING EQUATIONS

The Reynolds averaged Navier-Stokes equations in their conservative form is used. The equations are presented below in Cartesian coordinates.

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial \left(\rho_m u_j\right)}{\partial x_j} = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho_m u_i) + \frac{\partial}{\partial x_j}(\rho_m u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\mu + \mu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{3}$$

where the mixture density and turbulent viscosity are defined as

$$\rho_m = \rho_l \alpha_l + \rho_v \left(1 - \alpha_l\right) \tag{4}$$

$$\mu_t = \frac{\rho_m C_\mu k^2}{\varepsilon}$$

The k-ε turbulence model used is given by

$$\frac{\partial \rho_m k}{\partial t}+\frac{\partial \rho_m u_j k}{\partial x_j}=P-\rho_m \varepsilon+\frac{\partial\left[(\mu+\frac{\mu_t}{\sigma_k})\cdot\frac{\partial k}{\partial x_j}\right]}{\partial x_j} \tag{5}$$

$$\frac{\partial \rho_m \varepsilon}{\partial t}+\frac{\partial \rho_m u_j \varepsilon}{\partial x_j}=C_{\varepsilon 1}\frac{\varepsilon}{k}P-C_{\varepsilon 2}\rho_m\frac{\varepsilon^2}{k}+\frac{\partial\left[(\mu+\frac{\mu_t}{\sigma_\varepsilon})\cdot\frac{\partial \varepsilon}{\partial x_j}\right]}{\partial x_j} \tag{6}$$

where $C_{\varepsilon 1}$ regulates the production term and $C_{\varepsilon 2}$ regulates the dissipation term. The turbulent production term is defined as

$$P=\frac{\partial u_i}{\partial x_j}\tau_{ij} \tag{7}$$

where the Reynolds stresses are given by

$$\tau_{ij}=-\overline{\rho u_i' u_j'} \tag{8}$$

The volume fraction transport equation is given as

$$\frac{\partial \alpha_l}{\partial t}+\frac{\partial}{\partial x_j}(\alpha_l u_j)=(\dot{m}^+ + \dot{m}^-) \tag{9}$$

where the evaporation of the liquid phase is given by

$$\dot{m}^- = \frac{C_{dest}\rho_v \alpha_l MIN[0, p-p_v]}{\rho_l\left(\frac{1}{2}\rho_l U_\infty^2\right)t_\infty} \tag{10}$$

and the condensation of the vapor phase is given by

$$\dot{m}^+ = \frac{C_{prod}\rho_v \alpha_l^2 (1-\alpha_l)}{\rho_l t_\infty} \tag{11}$$

where C_{dest} and C_{prod} are the empirical parameters to be tuned depending on the flow. A value of 1000 is taken as the nominal density ratio, which is the ratio between thermodynamic values of density of liquid and vapor phases at the given flow conditions.

APPROACH

Flow Solver

The present Navier-Stokes solver employs a pressure-based algorithm and the finite volume approach [12, 14, 15]. The governing equations are solved on multiblock structured curvilinear grids in 2D and 3D domains. To represent the cavitation dynamics a transport equation model is adopted along with the particular source terms as suggested by Kunz et al[7]. For turbulent cavitating flow computations the pressure-based method of Senocak and Shyy[8] is used. One of the key features of this method is to reformulate the pressure correction equation to exhibit a convective-diffusive nature. This is achieved through the inclusion of the following pressure-velocity-density coupling scheme into the pressure correction equation.

$$\rho' = C\left(1-\alpha_l\right)P' \tag{12}$$

where C is an arbitrary constant of O(1). This scheme combines the incompressible and compressible formulations, so as to preserve the incompressibility of the liquid phase and it also takes into account the pressure-density dependency in the cavitating regions. Density is also upwinded both in pressure correction and momentum equations in order to improve the mass and momentum conservation in the vicinity of sharp density gradients. The convective terms of momentum and volume fraction transport equations are discretized using the second-order controlled variation scheme (CVS) [16, 17]. For more details of the pressure-based method for turbulent cavitating flows the reader is referred to Senocak and Shyy[8].

Sensitivity Analysis

The sensitivity of the objective function to the variation in the design parameters is evaluated with the aid of design of experiments and response surface methodology. Design of experiments is used to select the set of design variables, which are then used for CFD computations. These design variables are selected such that maximum amount of information about the design space can be obtained.

In this study for two design variables, design points are selected at equal intervals for each variable within the design space. For the four variables, full factorial design [18] is used to select the design points. Intervals of each design variable are divided into one or more subintervals, which mark the number of levels. For example, a two level design involves only the upper and lower limit of the design variables. A three level design variable will include an additional point in between. These points are usually evenly spaced. Full factorial design contains all the combinations of the levels of all the design variables.

A response surface (RS) is then generated using the CFD solutions for these design points. A polynomial-based RSM, in which the design space is represented with a quadratic polynomial, is used in this study. The polynomial coefficients are obtained by linear regression. The maximum or the minimum of the surface can then be located using a gradient search method. The RSM is effective in representing the global characteristics of the design space and it filters noise associated with design data. Depending on the order of polynomial employed and the shape of the actual response surface, the RSM can introduce substantial errors in certain regions of the design space. Generation of polynomial based surfaces can be costly for cases involving many design variables due to the amount of data required to evaluate the coefficients. In fact, the number of coefficients increases rapidly with the order of polynomial. For example, a complete second-order polynomial of N design variables has $(N+1)(N+2)/(2!)$ coefficients. A complete cubic model has $(N+1)(N+2)(N+3)/(3!)$ coefficients. The choice of order of the polynomial and the terms to be included depends on the design problem. The response surfaces were generated using JMP [18], a statistical analysis software package. JMP is an interactive, spreadsheet-based program having a variety of statistical analysis tools.

Once the response surface is generated it is submitted to an optimization toolbox to maximize/minimize the objective function. *Solver*[19], an optimization tool available as part of *Microsoft Excel* package, is used in this effort. This tool uses the *Generalized Reduced Gradient* (*GRG2*) nonlinear optimization code developed by Lasdon et al[20].

RESULTS AND DISCUSSION

Non-Cavitating Flow – Backward-Facing Step Flow

In flows with recirculation, the equilibrium between the turbulent production and dissipation does not exist. Hence a non-equilibrium model is used for the present computation. The solution of this model is compared with the solution obtained using the original k-ε turbulence model. Referring to eqns 4 and 5, the constants are defined as:

Model	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_k	σ_ε
Original k-ε	0.09	1.44	1.92	1.0	1.3
Non-equilibrium k-ε	0.09	$\alpha_1 + (1.40 - \alpha_1)(P/\varepsilon)$	$\beta_1 + (1.90 - \beta_1)(P/\varepsilon)$	0.8927	1.15

Table 1: Empirical constants used in k-ε turbulence model.

To estimate the sensitivities of α_1 and β_1 and hence $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$, flow over a backward facing step is considered. Geometry of length to step height ratio of 10 is used. The inlet to step height ratio is 2. A uniformly distributed grid of 121 × 91 is used. Calculations are carried out for Reynolds number of 10^6. Experimental results[21] predict the reattachment for such a flow as 7 ± 1. The comparison of CFD computations is made with a reattachment length of 7.

The design space for the two parameters, obtained from the turbulence model, α_1 and β_1 is selected based on past experience and the trend noticed in similar situations. The design space is as shown in the fig 1.

Computations are carried out on this backward facing step case with different values of α_1 and β_1. The reattachment length is obtained by tracking the grid point where the u-velocity becomes positive. This is done along the grid points closest to the bottom surface. The objective function for the sensitivity analysis is the absolute amount by which the reattachment length differs from a value of 7. Figure 2 gives the objective function as seen for the different combination of α_l and β_l. In fig 2, the circled values are for those designs which have a reattachment length within 7 ± 1.

The CFD simulations suggest that the best values of α_l and β_l lies within 10% variation on either side of the diagonal of the selected design space (Fig. 2). This agrees well with the observation that the turbulence model tries to maintain a balance between the production and dissipation of turbulence kinetic energy. The solutions also show that there are more than one optimum combination of α_l and β_l in the region mentioned.

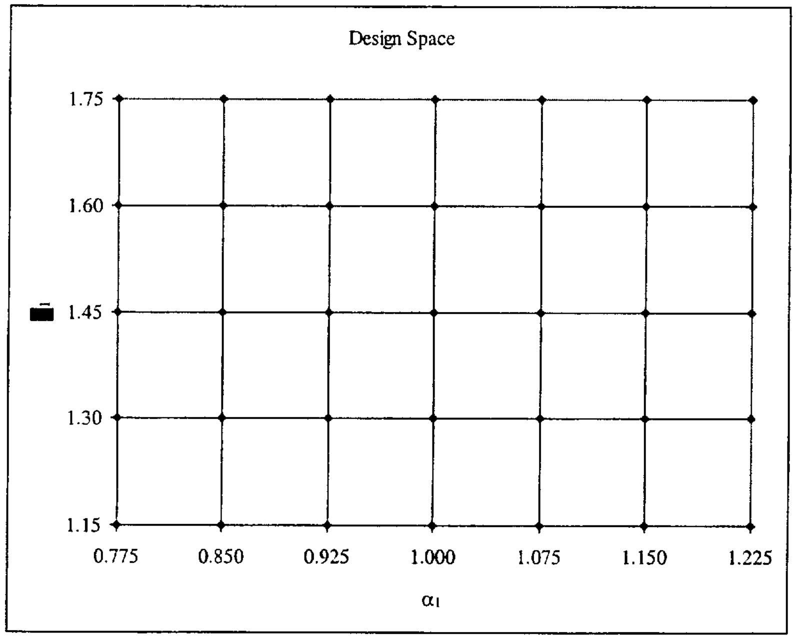


Figure 1: Design space for α_l and β_l.

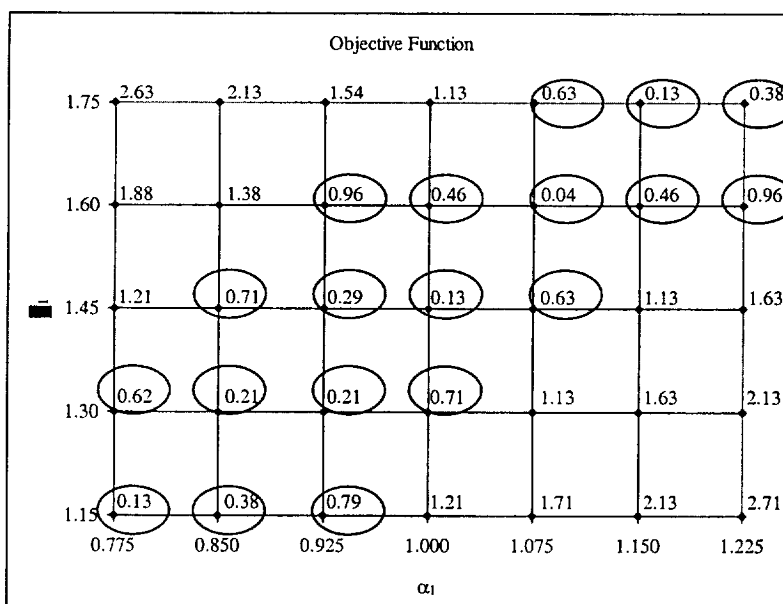


Figure 2: Objective function for different α_l and β_l.

A quadratic response surface is generated using the data obtained from CFD calculations that has the value of the objective function below 1. The response surface has an adjusted R^2 error of 0.916 and an estimated variance, σ_a, of 0.083. Viewing the plot between the predicted and actual values of the objective function, i.e., the error, it can be seen that the fit is good.

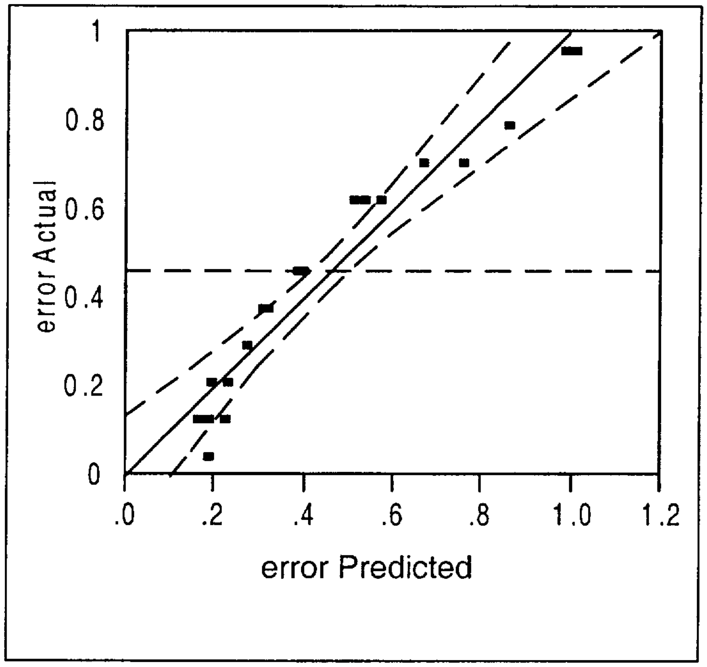


Figure 3: Comparison of response surface prediction of the objective function to the actual values.

In fig. 3, the dots represent the data. The continuous line represents the perfect model possible. The dashed lines represent the confidence limits pertaining to 95%. From the plot it is clearly seen that the fit is good as most of the points lie within the confidence limits. The values in the plot are not normalized.

The obtained response surface is

$$error = 0.4227 - 3.5038\alpha_1 + 1.9300\beta_1 + 36.1131\alpha_1^2 - 46.2841\alpha_1\beta_1 + 14.9684\beta_1^2 \quad (13)$$

In fig. 4, the values in the parenthesis are obtained from the response surface. They agree closely with the CFD solutions. Using this response surface a search for optimum values of α_1 and β_1 is carried out. The optimization problem is set up as follows:

Objective function: $error = |\text{CFD solution} - 7|$

Design variables: α_1, β_1

Constraints: $0.775 - \alpha_1 \leq 0$;

$\alpha_1 - 1.225 \leq 0$;

$1.150 - \beta_1 \leq 0$;

$\beta_1 - 1.750 \leq 0$.

The optimum value of the objective function is obtained for $\alpha_l = 0.7855$ and $\beta_l = 1.150$. The value of the objective function, *error*, is equal to 0.1583, as predicted by the response surface. CFD computation for the same values of the design parameters, give a value of 0.0417 for the *error*, which is the minimum among all the CFD runs made. The same value is also found at a different design point, suggesting that there are multiple optimum designs.

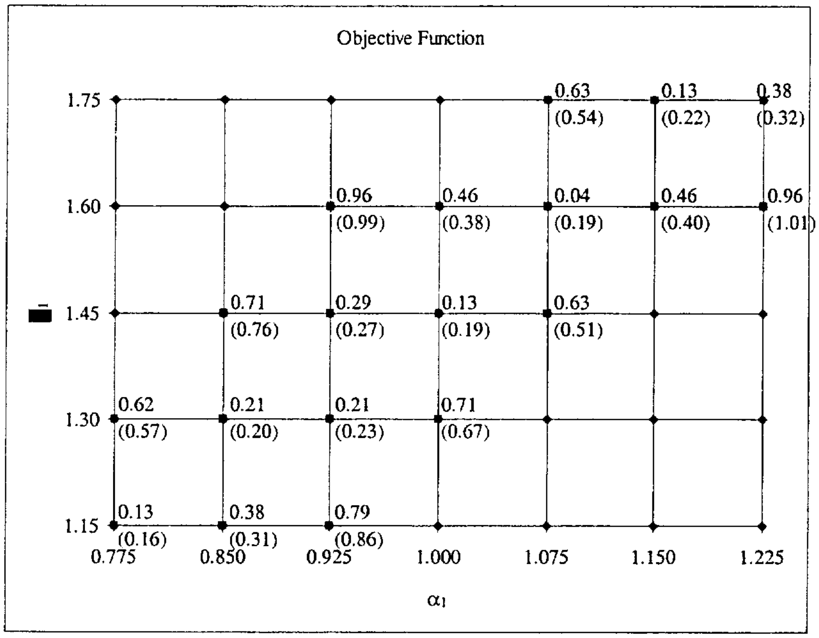


Figure 4: Comparison between CFD and response surface (in parenthesis) predictions of the objective function.

As noticed by this study, the best combination of α_l and β_l is one that balances the production and dissipation of turbulence kinetic energy. Hence the design space is reduced within a 10% variation on either side of the diagonal of the design space. It would save a lot of computation during the cavitating flow studies by concentrating on the reduced domain. Moreover there are multiple optimum values available in the domain and hence different choices of the modeling parameter will lead to the same value of the objective function.

Cavitating Flows

Computations for turbulent cavitating flow over two axisymmetric geometries have been performed. One of them has a hemispherical fore-body and the other has a blunt fore-body. Both the geometries have cylindrical aft-bodies. A steady state computation of cavitating flows is carried out with a Reynolds number of 1.36×10^5. As already mentioned the value of the cavitation parameter is different for the two geometries.

The performance of the cavitation model is evaluated by comparing the pressure coefficient on the surface of the body as obtained through CFD computations with those measured during experiments by Rouse and McNown[22]. As seen from eqns 9 and 10, two tuning parameters, namely, C_{dest} and C_{prod}, are involved. These parameters along with α_l

and β_l are the design parameters for this study. The sensitivities of these parameters are estimated from the cavitating flows over the mentioned geometries. A single block grid of 119×65 nodes is used for the hemispherical fore-body. For the blunt fore-body a grid system with two blocks having 55×55 and 255×85 nodes is used.

Hemispherical Fore-body

The upper and lower limits of the 4 design variables are selected based on experience.

Variable	Lower limit	Upper limit	Level
α_l	0.775	1.15	4
β_l	1.15	1.75	4
C_{dest}	8×10^5	1×10^6	3
C_{prod}	2×10^4	4×10^4	3

Table 2: Design space for the cavitating problem with hemispherical fore-body.

Using the full factorial design in the design of experiments, a design space of 144 design points is obtained. The design space is reduced to 81 with the aid of the trend noticed between α_l and β_l in the analysis of the backward-facing step case. The CFD computations are carried out for this set of design points. The objective function, *error*, based on the distribution of C_p along the surface of the cavitating body is measured as

$$error(C_p) = \frac{\sum_{1}^{12} |CFD - Expt|}{12} \qquad (14)$$

The comparison is done with the C_p values measured at 12 points along the surface of the body during experiments. The CFD computation using the original k-ε turbulence model along with the cavitation model parameters defined in Senocak and Shyy[8] (C_{dest}: 9×10^5 and C_{prod}: 3×10^4) will be considered as the baseline case. C_{dest} and C_{prod} values are normalized with respect to these baseline values. The cavitation number for all these CFD computations is 0.40. Based on the CFD computations the best and worst results obtained are:

Case	α_l	β_l	C_{dest} *(normalized)*	C_{prod} *(normalized)*	*error*
Best	0.90	1.15	0.889	1.333	0.0390
Worst	1.15	1.75	1.0	0.667	0.0699
Baseline			1.0	1.0	0.0420

Table 3: Best and worst results for the hemispherical fore-body based on comparison between CFD computations and experimental results.

From these results it can be seen that the best case lie at $\alpha_l = 0.9$ and $\beta_l = 1.15$. The error for the baseline case is 0.0420. Table 4 gives the cases, which have an *error* values within 10% of the best value. This shows that the best design points are located close to α_1 = 0.9 and β_1 = 1.15. Figure 5 shows the sensitivity of C_{dest} and C_{prod} for α_1 = 0.9 and β_1 = 1.15. It shows that the best cases are towards the lower limits of C_{dest} and upper limits of C_{prod} of the chosen domain. Figure 6 shows the distribution of pressure coefficient as suggested by the best solution from CFD computations. The results confirms well with the experimental measurements. There is slight discrepancy at the closure region of the cavity.

α_1	β_1	C_{dest} *(normalized)*	C_{prod} *(normalized)*	*error*
0.9	1.15	0.889	1.333	0.0390
0.775	1.15	0.889	1.333	0.0399
0.9	1.15	0.889	1.0	0.0410
0.9	1.15	1.111	1.333	0.0412
0.9	1.35	0.889	1.333	0.0418
1.025	1.35	0.889	1.333	0.0422
0.775	1.15	1.0	0.667	0.0425
0.775	1.35	1.0	1.0	0.0427
0.9	1.15	1.151.0	1.333	0.0427
1.15	1.75	1.0	1.333	0.0427

Table 4: Cases within 10% variation of *error* as compared to the best case. (i.e. within *error* = 0.0428).

Based on the CFD results, a quadratic response surface is generated using the Iteratively Re-weighted Least Square method. The plot (fig. 7) between the predicted and the actual value of *error*, suggest a good fit.

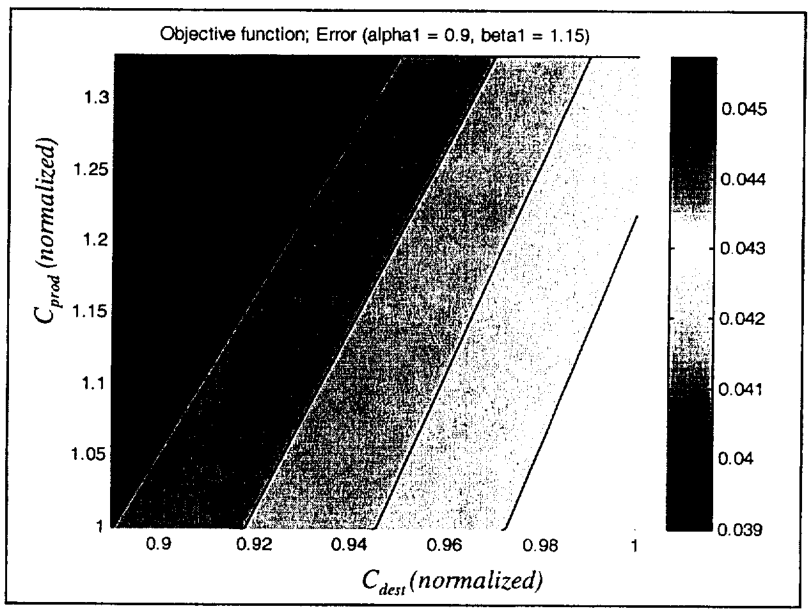


Figure 5: Sensitivity of C_{dest} and C_{prod} for α_1 = 0.9 and β_1 = 1.15 for the hemispherical fore-body.

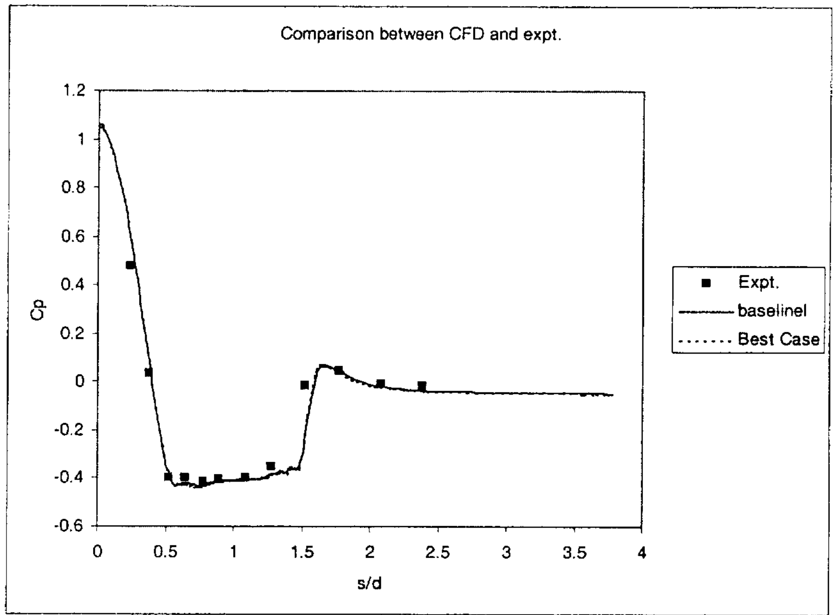


Figure 6: Comparing the C_p values for the best CFD case with the CFD baseline and experimental results for the hemispherical fore-body.

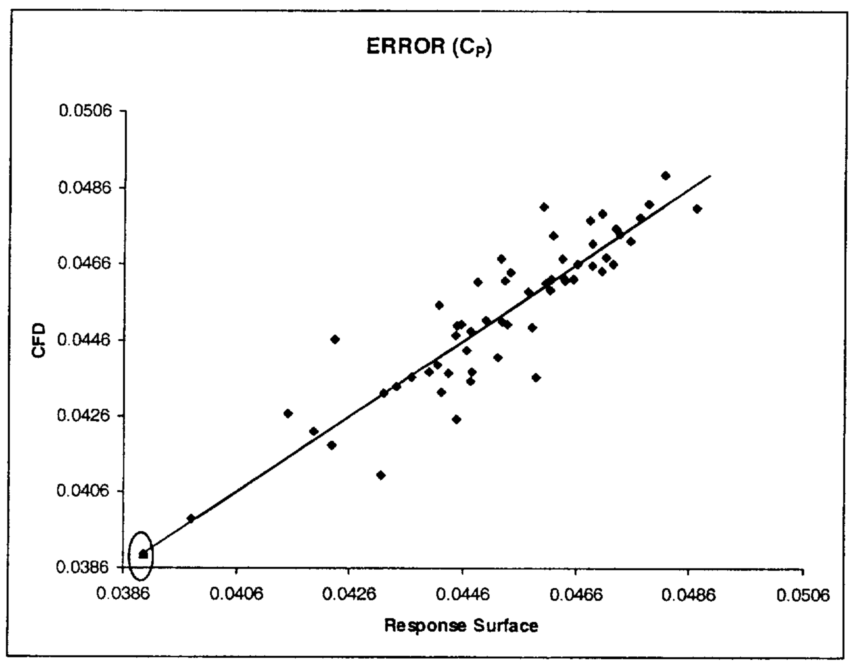


Figure 7: Comparison of response surface prediction of the objective function to the actual values.

In fig. 7, the line indicates the perfect model. The distribution of data around the ideal model suggests a good fit. The best fit is circled. This is identified accurately by both the response surface and CFD computations. In table 5, the optimum as predicted by the response surface is shown in the second row. Third row shows the best case as noticed from the CFD runs. These two identify different optimum combinations of the parameters but in the near vicinity of each other.

α_l	β_l	C_{dest} *(normalized)*	C_{prod} *(normalized)*	*error (CFD)*	*error (RS)*
1.012	1.15	0.889	1.333	0.0411	0.0387
0.9	1.15	0.889	1.333	0.0390	0.0389

Table 5: Optimum values of the parameters obtained from response surface and CFD computations.

To check how the sensitivity of these design parameters change with the cavitation number, few CFD runs were done with a cavitation number of 0.3. As seen from table 6, the performance is good as compared to the *error* value of 0.0548 obtained for the baseline case with a cavitation number of 0.3. It is also seen that $\alpha_l = 0.9$ and $\beta_l = 1.15$ gives good performance, which suggests that this is a good choice for the turbulence modeling parameters.

α_l	β_l	C_{dest} *(normalized)*	C_{prod} *(normalized)*	*error*
0.9	1.15	0.889	1.333	0.0504
0.775	1.15	0.889	1.333	0.0502
0.9	1.15	0.889	1.0	0.0524
0.9	1.15	1.111	1.333	0.0520

Table 6: CFD computations for the hemispherical fore-body with cavitation number equal to 0.3.

Blunt Fore-body

Based on the results obtained from the hemispherical fore-body studies, the CFD computation on blunt fore-body is done for $\alpha_l = 0.9$ and $\beta_l = 1.15$. The values of C_{dest} and C_{prod} are varied as shown in table 7.

Variable	Lower limit	Upper limit	Level
C_{dest}	0.8×10^4	2.8×10^4	3
C_{prod}	5.0×10^3	7.0×10^3	3

Table 7: Design space for cavitating problem with blunt fore-body.

This gives 9 design points. The baseline case used for comparison has C_{dest} (normalized): 0.02 and C_{prod} (normalized): 0.2 along with the original k-ε turbulence model. The same measure of *error* is used for comparison in this case. In this case the number of experimental points available is 20. The cavitation number for these cases is 0.5.

Based on the CFD computations the best and worst results obtained are:

Case	α_1	β_1	C_{dest} *(normalized)*	C_{prod} *(normalized)*	*error*
Best	0.90	1.15	0.0312	0.167	0.0907
Worst	0.90	1.15	0.0089	0.234	0.1608

Table 8: Best and worst results for the blunt fore-body based on comparison between CFD computations and experimental results.

Figure 8 shows the sensitivity of C_{dest} and C_{prod} for $\alpha_1 = 0.9$ and $\beta_1 = 1.15$ in the case of the blunt fore-body. It shows that the best cases are towards the upper limits of C_{dest} and lower limits of C_{prod} of the chosen domain.

Comparing the distribution of pressure coefficient along the surface of the body, gives an idea of the performance of the models. Figure 9 compares the performance between the best CFD case mentioned in table 8 with the CFD case using the original k-ε turbulence model and the experimental results. The best case performs better than the baseline case, although there is a slight discrepancy as compared to the experimental results. This shows that the CFD computation can benefit if modeling parameters are appropriately selected.

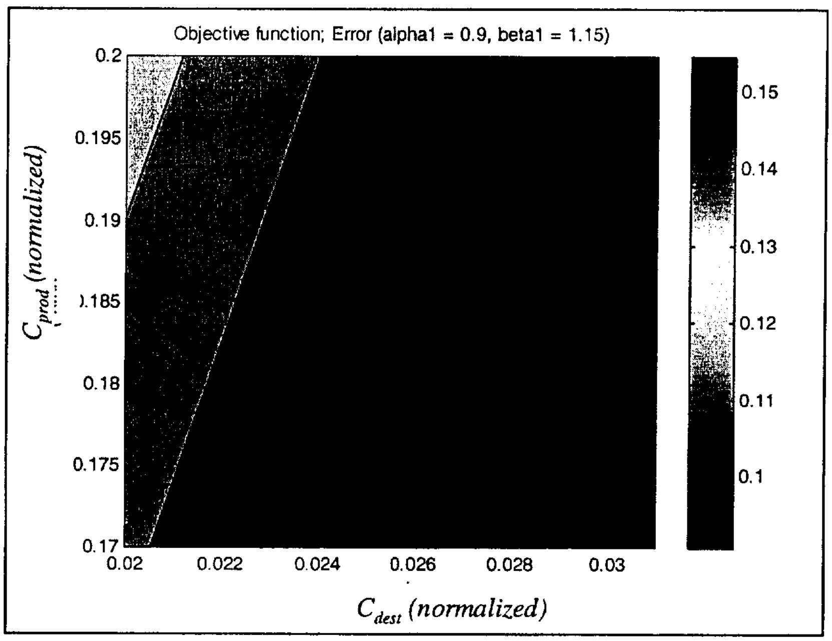


Figure 8: Sensitivity of C_{dest} and C_{prod} for $\alpha_1 = 0.9$ and $\beta_1 = 1.15$ for the blunt fore-body.

CFD computations are done on the hemispherical fore-body with the design parameters of the best CFD case of the blunt fore-body. Figure 10 shows the results. With these values of C_{dest} and C_{prod}, the evaporation and condensation of the phases are not balanced. The CFD computations have a problem converging. This suggests that the steady state assumptions may not be valid for this case. The cavitation number is 0.4. Using a cavitation number of 0.5 results in the same problem. Further study has to be done to address the unsteady cavitating model and the influence of the geometry of the body on cavitation.

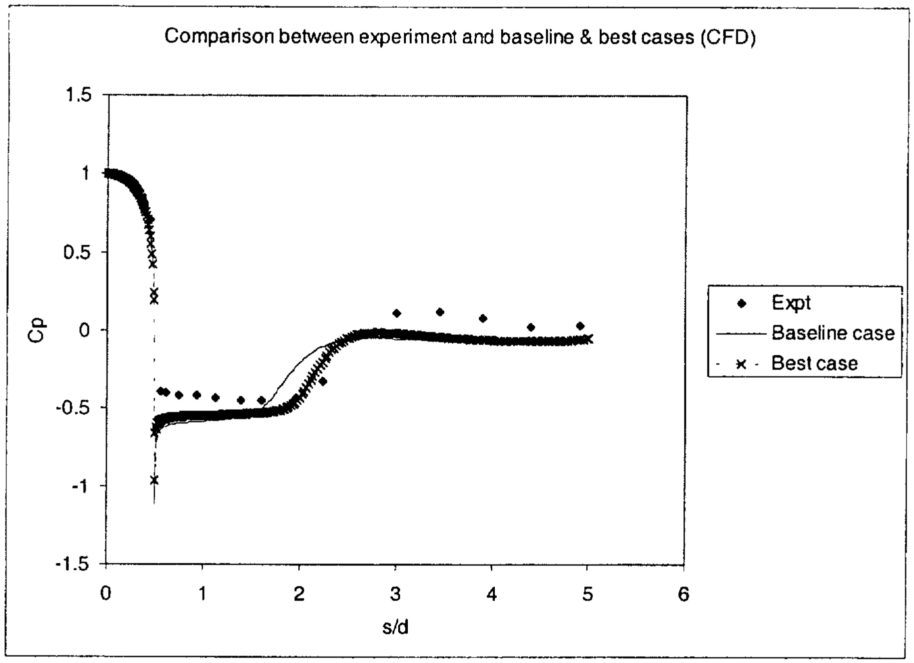


Figure 9: Comparing the C_p values for the best CFD case with the CFD baseline and experimental results for the blunt fore-body.

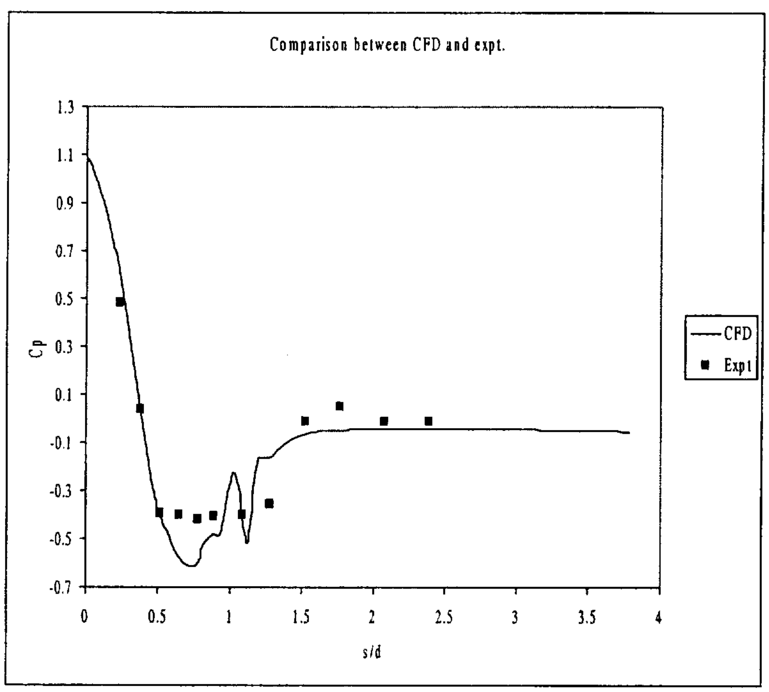


Figure 10: Comparing the C_p values for the CFD computation on the hemispherical fore-body (using the best parameters of the blunt fore-body) and experimental results.

CONCLUSION

This study gives an insight into the trend of the different important parameters of turbulence and cavitation models. It clearly defines the region to concentrate, in terms of their ideal values.

The backward facing step case clearly points out the trend of the turbulence parameters. It is as expected based on theoretical arguments. The parameters $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$, regulate the production and dissipation of turbulent kinetic energy respectively. Hence when one increases the other has to increase. This is clearly seen from the computational results where an almost linear relationship between α_1 and β_1 is noticed.

The cavitating cases offer a clear picture into the other design parameters encountered in their modeling. The cavitation model for hemispherical fore-body performs better with low values of C_{dest} and high values of C_{prod} within the selected design space. The cavitating model for the blunt fore-body performs better with high values of C_{dest} and low values of C_{prod} within the selected design space.

The convergence for the cavitating cases requires different values of C_{dest} and C_{prod}. The values required for the hemispherical fore-body geometry are an order higher than those required for the blunt fore-body geometry. This suggests that these parameters maybe highly case dependent.

This study gives a method for selecting the tuning parameters for non-cavitating and cavitating flows, which can improve the performance. This also brings to notice, the fact that unsteady issues needs to be addressed to improve the performance. Future studies intend to come up with a general scheme that may be able to aid the modeling of identical flows over geometries.

REFERENCES

1. Trevena, D. H. (1987), *Cavitation and Tension in Liquids*, Adam Hilger, Bristol and Philadelphia.
2. Brennen, C. E. (1995), *Cavitation and Bubble Dynamics*, Oxford University Press, New York.
3. Knapp, R. T., Daily, J. W. and Hammitt, F. G. (1970), *Cavitation*, McGraw-Hill, New York.
4. Lecoffre, Y. (1999), *Cavitation Bubble Trackers*, A. A. Balkema, Brookfield, VT.
5. Singhal, A. K., Vaidyan, N. and Leonard, A. D. (1997), "Multi-dimensional Simulation of Cavitating Flows Using a PDF Model for Phase Change," ASME Paper FEDSM97-3272, the 1997 ASME Fluids Engineering Division Summer Meeting.
6. Merkle, C. L., Feng, J. and Buelow, P. E. O. (1998), "Computational Modeling of the Dynamics of Sheet Cavitation," 3rd International Symposium on Cavitation, Grenoble, France.
7. Kunz, R. F., Boger, D. A., Stinebring, D. R., Chyczewski, T.S., Lindua, J. W., Gibeling H. J., Venkateswaran, S. and Govindan, T. R. (2000), " A pre-conditioned Navier-Stokes Method for Two-phase Flows with Application to Cavitation Prediction," *Computers & Fluids*, Vol. 29, pp. 849-875.
8. Senocak, I. And Shyy, W. (2001), "A pressure-Based Method for Turbulent Cavitating Flow Computations," 31st AIAA Fluid Dynamics Conference and Exhibit, AIAA 2001-2907.

9. Senocak, I. and Shyy, W. (2001), "Numerical Simulation of Turbulent Flows with Sheet Cavitation," CAV2001 4th International Symposium on Cavitation, Paper No. AV2001A7.002.
10. Ahuja, V., Hosangadi, A. and Arunajatesan, S. (2001), "Simulations of Cavitating Flows Using Hybrid Unstructured Meshes," *Journal of Fluids Engineering*, Vol. 123, No. 2, pp. 331-340.
11. Patankar, S. V. (1980), *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington DC.
12. Shyy, W., Thakur, S. S., Ouyang, H., Liu, J. and Blosch, E. (1997), *Computational Techniques for Complex Transport Phenomena*, Cambridge University Press, New York.
13. Myers, R. H. and Montgomery, D. C. (1995), *Response Surface Methodology – Process and Product Optimization Using Designed Experiment*, John Wiley & Sons.
14. Shyy, W. (1994), *Computational Modeling for Fluid Flow and Interfacial Trasport*, Elsevier, Amsterdam, The Netherlands, (revised printing 1997).
15. Thakur, S. S., Wright, J. F., Shyy, W. and Udaykumar, H. (1997), "SEAL: A Computational Fluid Dynamics and Heat Transfer Code for Complex 3-D Geometries," University of Florida, Gainesville.
16. Shyy, W. and Thakur, S. S. (1994), "A Controlled Variation Scheme in a Sequential Solver for Recirculating Flows. Part I. Theory and Formulation," *Numer. Heat Transf.*, Part B, Vol. 25, No. 3, pp. 245-272.
17. Shyy, W. and Thakur, S. S. (1994), "A Controlled Variation Scheme in a Sequential Solver for Recirculating Flows. Part II. Applications," *Numer. Heat Transf.*, Part A, Vol. 25, No. 3, pp. 273-286.
18. SAS Institute Inc. (1995). *JMP version 3.* Cary, NC.
19. Microsoft Corporation. (1985-1999). *Microsoft Excel 2000.*
20. Lasdon, L.S, Waren, A., Jain, A. and Ratner, M. (1978), "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," ACM Transactions on Mathematical Software 4:1.
21. Kim, J., Kline, S. J. and Johnston, J. P. (1978), Investigation of Separation and Reattachment of a Turbulent Shear Layer: Flow over a Backward-Facing Step, Report No. MD-37, Thermosciences Division, Dept. of Mechanical Engineering, Standford University.
22. Rouse, H. and McNown, J. S. (1948), "Cavitation and Pressure Distribution, Head Forms at Zero Angle of Yaw," Studies in Engineering, Bulletin 32, State University of Iowa.

PERGAMON

Progress in Aerospace Sciences 37 (2001) 59–118

PROGRESS IN AEROSPACE SCIENCES

www.elsevier.com/locate/paerosci

Global design optimization for aerodynamics and rocket propulsion components

Wei Shyy[a,*], Nilay Papila[a], Rajkumar Vaidyanathan[a], Kevin Tucker[b]

[a]*Department of Aerospace Engineering, Mechanics and Engineering Science, University of Florida, 231 Aerospace Building, Gainesville, FL 32611-2031, USA*
[b]*NASA Marshall Space Flight Center, AL, USA*

Abstract

Modern computational and experimental tools for aerodynamics and propulsion applications have matured to a stage where they can provide substantial insight into engineering processes involving fluid flows, and can be fruitfully utilized to help improve the design of practical devices. In particular, rapid and continuous development in aerospace engineering demands that new design concepts be regularly proposed to meet goals for increased performance, robustness and safety while concurrently decreasing cost. To date, the majority of the effort in design optimization of fluid dynamics has relied on gradient-based search algorithms. Global optimization methods can utilize the information collected from various sources and by different tools. These methods offer multi-criterion optimization, handle the existence of multiple design points and trade-offs via insight into the entire design space, can easily perform tasks in parallel, and are often effective in filtering the noise intrinsic to numerical and experimental data. However, a successful application of the global optimization method needs to address issues related to data requirements with an increase in the number of design variables, and methods for predicting the model performance. In this article, we review recent progress made in establishing suitable global optimization techniques employing neural-network- and polynomial-based response surface methodologies. Issues addressed include techniques for construction of the response surface, design of experiment techniques for supplying information in an economical manner, optimization procedures and multi-level techniques, and assessment of relative performance between polynomials and neural networks. Examples drawn from wing aerodynamics, turbulent diffuser flows, gas–gas injectors, and supersonic turbines are employed to help demonstrate the issues involved in an engineering design context. Both the usefulness of the existing knowledge to aid current design practices and the need for future research are identified.

Keywords: Global optimization; Response surface methodology; Design of experiments; Neural networks

Contents

* Corresponding author. Tel.: +1-352-392-0961; fax: +1-352-392-7303.
E-mail address: wei-shyy@ufl.edu (W. Shyy).

PII: S0376-0421(01)00002-1

1. Introduction and scope

Modern computational and experimental fluid dynamics tools have matured to a stage where they can provide substantial insight into engineering processes involving fluid flows. This can help analyze the fluid physics as well as improve the design of practical devices. In particular, rapid and continuous development in the technology of fluid machinery demands that new design concepts be regularly proposed to meet goals for increased performance, robustness and safety while concurrently decreasing cost.

Most aerospace system and component designs are conducted as open-loop, feed-forward processes. For example, for rocket engines, currently, one design iteration for a given set of engine balance conditions takes up to several weeks with the blade geometry design sub-iteration phase taking several days each. The quest for an acceptable blade surface velocity distribution is accomplished with many ad hoc rules in what is essentially a manual trial-and-error process. A systematic approach capable of identifying optimum design and comparing possible trade-offs can significantly improve the productivity and shorten the design cycle.

Objective and efficient evaluation of advanced designs can be facilitated by development and implementation of systematic optimization methods. To date, the majority of the effort in design optimization of fluid dynamics has relied on gradient-based search algorithms [1–3]. These methods work iteratively through a sequence of local sub-problems, which approximate objective and constraint functions for a sub-region of the design space, e.g., by linearization using computed sensitivities. Major challenges for these optimization approaches are the robust and speedy computation of sensitivity coefficients [4,5].

Local optimization methods based on derivatives are also commonly used in engineering system design optimization problems [6]. On the other hand, global optimization techniques also have been commonly used for engineering design optimization problems especially for multidisciplinary ones. In its current practice, the global design optimization method involves three primary steps (Fig. 1): (a) generation of individual data sets within the design space; (b) interpolation among these data sets via some continuous functional representation; and (c) optimization of the objective function via a certain search strategy. Yet despite recent research advances, formal design optimization has yet to see practical use in real design scenarios. The reasons are four-fold:

(1) Engineering design, even within a single discipline, typically involves many parameters (and hence many degrees of freedom) rather than the handful demonstrated in most research papers. This renders unrestricted "brute force" search schemes too resource-intensive.
(2) The objective functions are likely to be multi-modal or discontinuous over the broad design space, rendering gradient search methods insufficient by themselves. Furthermore, the usual practice to combine multiple goals into a single quantitative objective function is too restrictive. *Qualitative* goals are often required to correctly characterize a problem (e.g., maximizing a turbine blade's aerodynamic efficiency with a *smooth, monotonic* surface velocity distribution, while spreading heat load as *uniformly as possible*). Furthermore, these goals may have arisen from diverse disciplines and are usually treated sequentially by different groups.
(3) It is inadequate to think of the final product of a design process as a mere geometry. A "design" really encompasses a whole set of operating, manufacturing and project level decisions.
(4) As the interaction between numerical simulation and physical test data becomes stronger, the future engineering knowledge base is likely to consist of all

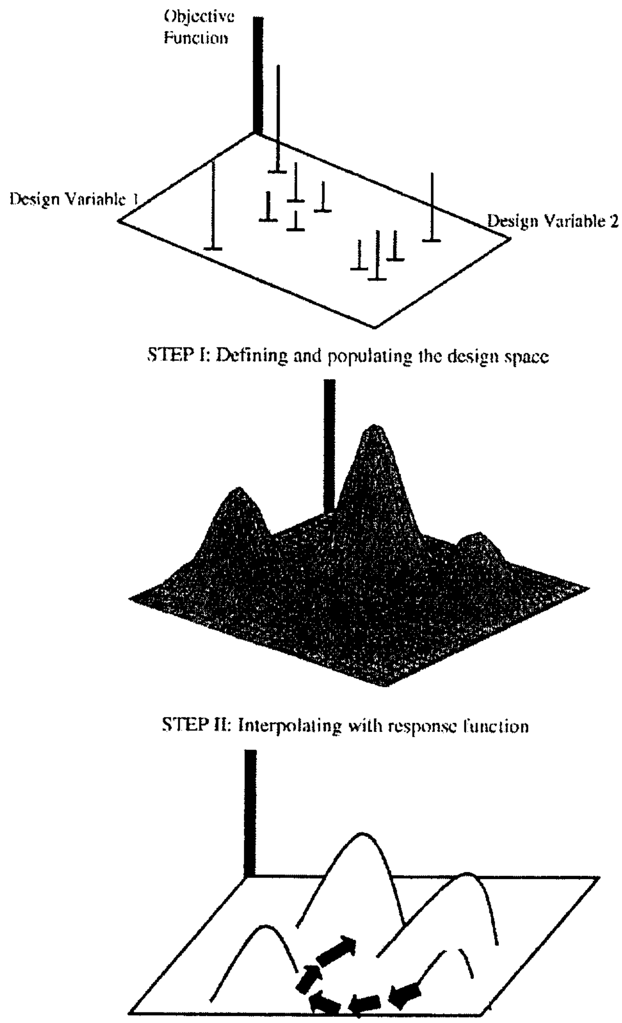


Fig. 1. Schematic of the procedure for global design optimization.

sorts of heterogeneous data sources including test data, experimental data, past product experiences, semi-empirical modeling, and high fidelity simulations. Some data are anecdotal; others cover only small "patches" of the physical domain but are still useful for "reality checks". A unified framework needs to be constructed for representation, capturing and mining of all these data types so the response functions can be continuously improved.

With the above observations, global optimization methods are attractive because they have several advantages when compared to local gradient-based methods [7]:

(1) They do not require calculation of the local sensitivity of each design variable,
(2) They can utilize the information collected from various sources and by different tools,
(3) They offer multi-criterion optimization,
(4) They can handle the existence of multiple design points and trade-offs,
(5) They easily perform tasks in parallel, and
(6) They can often effectively filter the noise intrinsic to numerical and experimental data.

Among global approximation techniques, the *response surface methodology* (RSM) has gained the most attention since it consists of a simple way of connecting codes from various disciplines [6]. The RSM is a collection of mathematical and statistical tools used in investigative experimentation by scientists and engineers [8]. The RSM approach replaces the objective and constraint functions by simple functions, often polynomials, which are fitted to the carefully selected points. Since RSM can utilize information collected from various sources and by different tools, it can also offer multi-criterion optimization, handle the existence of multiple design selections and related trade-offs, and address the noises intrinsic to numerical and experimental data. A main advantage of RSM is its robustness and intelligibility. Robustness and the smoothness of approximations in noisy environments are achieved by performing extra analyses, compared to the number of regression coefficients. This is a distinct advantage over derivative-based search algorithms, which may encounter difficulties in the presence of spurious local optima [9].

1.1. Scope

In this article, we first review the basic concepts and methodologies, then assess the current status, via case studies, of the global optimization techniques. Particular attention is paid to two different techniques used to generate information to construct the *response surface* (RS) namely; *neural-network* (NN)- and polynomial-based RSM. NNs are models that contain many simple linear and non-linear elements operating in parallel and connected in patterns [10]. Polynomial-based RSM models the system with polynomials of assumed order and unknown coefficients. The solution for the set of coefficients that best fits the training data is a linear least-squares problem, making it trivial compared to the solution for NN, which often involves a non-linear training process. In this article, two neural network types, namely, *back-propagation* NN (BPNN) and *radial basis* NN (RBNN), are investigated.

The BPNN consists of multi-layer networks with differentiable activation function. The BPNN is the most employed NN type in the optimization literature [10–33].

The RBNN is a more recently developed multi-layer network, based on a linear regression process, which makes the mathematics simpler and computational costs lower [34–36]. The RBNN tends to have many more neurons than BPNN but can be configured faster for the same training data. The basic reason for this is that back-propagation neurons can have outputs over a large

Table 1
Comparison of NN and polynomial-based response surface (RS) techniques

	NN-based RSM	Polynomial-based RSM	Comments
Computational effort and cost	Disadvantage	Advantage	Finding the weights associated with the neurons is a non-linear regression process for all of the NN types other than RBNN. Whereas, finding the polynomial coefficients requires solution of a linear set of equations The cost increases if the regression process is non-linear which makes NNs other than RBNN more expensive than polynomials
Noise	Disadvantage	Advantage	Ability of filtering noise from experimental data is possible with polynomial-based RSM. However, if the number of neurons used to design the NN is not the same as the data, then, by definition, filtering is also possible for NN-based RSM
Handling complex functions	Advantage	Disadvantage	NNs are more suitable for multi-dimensional interpolation of data that lack structure since they are much more flexible in functional form especially when dealing with design in the context of unsteady flows, partial and/or complete data sets

region of the input space, while radial-basis neurons respond to relatively small regions of the input space. Thus, larger input spaces require more radial-basis neurons for training. More detailed evaluation of RBNN and BPNN will be given in the following sections.

Polynomial-based response surfaces and linear regression techniques were originally developed to filter noise from experimental data. Sophisticated statistical tools are available for these purposes. One class of tools, *design of experiments*, is often used to select points for training that minimize the effect of noise on the fitted polynomial. A second set of tools, *analysis of variance*, is routinely used to identify polynomial coefficients that are not well characterized by the data, and are therefore overly sensitive to noise. Analysis of variance helps to avoid overfitting of the data, which otherwise would result in the mapping of the noise. On the other hand, neural networks are much more flexible in functional form, which means that they can be better suited to fit complex functions that are not easily approximated by polynomials. For example, when the physical system changes from one regime to another due to the presence of critical parameters, NN performs better than RSM. This advantage is particularly useful when there is very little numerical noise, and it is possible to obtain very accurate approximations to the underlying function [37]. The relative strengths and weaknesses of NN- and polynomial-based RSM are summarized in Table 1.

Table 2 summarizes the existing literature evaluating the relative performance of NN- and polynomial-based RSM approximation. For example, Carpenter and Barthelemy [11] used NN- and polynomial-based approximations to develop RS for several test problems. It is demonstrated that two methods perform comparably based on the number of undetermined parameters. Rai and Madavan [27] investigated the feasibility of applying neural networks to the design of turbomachinery airfoils. The NN approach is used for both function approximation and prediction. It is found that NNs are quite efficient in both tasks. An aerodynamic design procedure that employs a strategy called parameter-based partitioning incorporating the advantages of both traditional RSM and NNs to create a composite response surface is described by Rai and Madavan [28,29]. It is shown that such method can handle design problems with higher-dimensional problems than would be possible using NN alone. Nikolaidis et al. [25] used NNs and response surface polynomials to predict the performance characteristics of automotive joints using geometrical parameters. It is shown that both methods performed comparably. NN-based aerodynamic design procedure is applied to the redesign of a transonic turbine stage to improve its unsteady aerodynamic performance by Madavan et al. [22]. It is illustrated that using an optimization procedure combining the advantages of NN- and polynomial-based RSM can be advantageous. Papila et al. [37] investigated the relative merits of polynomial-based RSM, RBNN and BPNN in handling different data characteristics. It is demonstrated that using RBNN rather than BPNN has certain advantages as data size increases. Also, it is shown that RBNN gives more accurate results than polynomial-based RSM as the nature of the experimental data becomes complex. Shyy et al. [38] have employed neural network techniques and polynomial-based RSM to obtain improved optimization tools. In Rai and Madavan [29], a composite NN and polynomial-based RS methodology is applied for a transonic turbine and it

Table 2
Literature review on NN and polynomial-based RS techniques comparison

Authors	No. of data	No. of input	No. of output	NN-type (2-layer)	Activation function	No. of neurons		Polynomial degree
Carpenter and Barthelemy [11]	36	2	1	BPNN	Sigmoid	1, 2, 4		1–4
	961	2	1			3, 5, 7		2–5
	81	4	1			1, 2, 3		1–2
	300	15	1			2, 4, 6, 8, 10		1–2
								1–2
Madavan et al. [22]	—	13	1	BPNN (3-layer)	Sigmoid	15 & 7		1–2
Nikolaidis et al. [25]	400	50	1	BPNN	Sigmoid	σ of NN is insensitive to no. of neurons		2
Papila et al. [37]	9	2	1	RBNN & BPNN	Radbas & Sigmoid	8, 9	4	2–5
	15	2	1			12, 15	4	2–5
	25	2	1			20, 25	4	2–5
	255	2	1			253, 255	—	2–4
	765	2	1			765	—	—
Rai and Madavan [28]	3 & 5	1	1	BPNN (3-layer)	Sigmoid	1 & 2		1–2
	27	3	1			7 & 3		1–2
	—	15	1			—		—
Shyy et al. [38]	45	3	2	RBNN	Radbas	42 and 45		2–3
Vaidyanathan et al. [39]	45	3	2	RBNN	Radbas	42 and 45		2–3
	76	6	2					

is demonstrated that a systematic application of such method can enhance the effectiveness of the overall optimization process. In the study by Vaidyanathan et al. [39], the application of NN- and polynomial-based RSM in preliminary design of two rocket engine components, gas–gas injector and supersonic turbine, with modest amounts of data are discussed and it is demonstrated that NN- and polynomial-based approximations can perform comparably for modest data sizes.

In this article, we focus on the recent efforts in developing and improving appropriate techniques for design optimization of airfoils and rocket engine components capable of being used in applications like *reusable launch vehicles*. Some of the physical components used as case studies are low Reynolds number aerodynamics, 2-D turbulent planar diffuser, the injector and the supersonic turbine for rocket propulsion.

Specifically, the following issues are discussed:

(1) The capability of the NN- and polynomial-based RSM for handling data with variable sizes and noise.
(2) The selection of NN configuration that is suitable for given design problems.
(3) The effect of the design parameters on the performance of the NN.
(4) The effect of distribution of the data over the design space in the construction of the global model.
(5) The merit of employing a multi-level optimization strategy to perform the task adaptively and efficiently.
(6) Possible trade-offs between capacity design objectives and their impact on design selections.

2. Review of methodologies

In response-surface-based global optimization, there are several key technical elements:

(1) Response surface with polynomials and statistical analysis.
(2) NNs with BPNN and RBNN.
(3) Design of experiments with *face centered composite design* (FCCD), *orthogonal arrays* (OA) and D-optimal designs.
(4) Optimization procedure including the multilevel approach.

In the following, we review these elements in sequence.

2.1. Response surface method (RSM)

The approach of RSM is to perform a series of experiments, based on numerical analyses or physical experiments, for a prescribed set of design points, and to construct a global approximation of the measured quantity over the design space (Fig. 1). The polynomial-based RSM, used in all the case studies referred to, construct polynomials of assumed order and unknown coefficients based on regression analysis. The number of coefficients to be evaluated depends on the order of polynomial and the number of design parameters involved. For instance, a second-order polynomial of N design variables has $(N+1)(N+2)/(2!)$ coefficients. A cubic model has $(N+1)(N+2)(N+3)/(3!)$ coefficients. In this article, the polynomial approximations are constructed by standard least-squares regression using JMP [40], a statistical analysis software that provides a variety of statistical analysis functions in an interactive format.

In the practical application of RSM, it is necessary to develop an approximate model for the true response surface. The second-order (quadratic) response surface model is the most frequently applied one because it is the most economic non-linear model. Such a model for response variable y with k regressors can be written as

$$\mathbf{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=2}^{k} \beta_{ij} x_i x_j + \boldsymbol{\varepsilon}. \quad (1)$$

The above equation can be written in matrix notation as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (2)$$

where y is the $(n \times 1)$ vector of observations, $\mathbf{X}$ the $(n \times n_p)$ matrix of the levels of the independent variables, $\boldsymbol{\beta}$ the $(n_p \times 1)$ vector of the regression coefficients, $\boldsymbol{\varepsilon}$ the $(n \times 1)$ vector of random error, n the number of observations, and n_p the number of terms in the model.

The purpose is to find the vector of least-squares estimators, $\mathbf{b}$, that minimizes

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad (3)$$

which yields to the least-squares estimator of $\boldsymbol{\beta}$

$$\mathbf{b} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \quad (4)$$

The global fit and prediction accuracies of the response surfaces are assessed through statistical measures such as the t-statistic, or t-ratio, root-mean-square error (rms-error), variation [41]. The t-statistic is determined by

$$t = \frac{b_j}{\mathrm{se}(b_j)}, \quad (5)$$

where b_j is the least-squares estimators of the jth regression coefficient and $\mathrm{se}(b_j)$ is the standard error of b_j and it is given by

$$\mathrm{se}(b_j) = \sigma_{\mathrm{a}}\sqrt{C_{jj}}, \quad (6)$$

where C_{jj} is the diagonal element of $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ corresponding to b_j. Here σ_{a} is the adjusted rms-error (or rms-error predictor) incurred while mapping the surface over the data set. The quality of the fit of the different surfaces can be evaluated by comparing the adjusted rms-error value that is defined as

$$\sigma_{\mathrm{a}} = \sqrt{\frac{\sum e_i^2}{n - n_p}}, \quad (7)$$

where e_i is the error at ith point of the training data.

The accuracy of the models in representing the design space is gauged by comparing the values of the objective function at test design points, different from those used to generate the fit, with the empirical solution. The prediction rms-error, σ, for the test set is given by

$$\sigma = \sqrt{\frac{\varepsilon_i^2}{m}}. \quad (8)$$

In this equation ε_i is the error at the ith test point and m is the number of test points.

The coefficient of multiple determination R^2 measures the proportion of the variation in the response around the mean that can be attributed to terms in the model rather than to random error and it is determined by

$$R^2 = \frac{\mathrm{SS}_R}{\mathrm{SS}_{yy}} = 1 - \frac{\mathrm{SS}_E}{\mathrm{SS}_{yy}}, \quad (9)$$

where SS_E is the sum of squares of the residuals $(=\sum_{i=1}^{n}(y_i - \hat{y}_i)^2)$ where $\hat{\mathbf{y}}$ is the predicted value by the fitted model. SS_R is the sum of squares due to regression $(=\sum_{i=1}^{n}(\hat{y}_i - \bar{y}_i)^2)$ where $\bar{\mathbf{y}}$ is the overall average of y_i. SS_{yy} is the total sum of squares about the mean given by

$$\mathrm{SS}_{yy} = \mathrm{SS}_E + \mathrm{SS}_R = \sum_{i=1}^{n}(y_i - \bar{y})^2, \quad (10)$$

where $\bar{\mathbf{y}}$ is the overall average of y_i.

R_{a}^2 is an R^2 value adjusted to account for the degrees of freedom in the model and is given by

$$R_{\mathrm{a}}^2 = 1 - \frac{\mathrm{SS}_E/(n-p)}{\mathrm{SS}_{yy}/(n-1)} = 1 - \left(\frac{n-1}{n-p}\right)(1 - R^2). \quad (11)$$

Since R^2 increases as terms are added to the model, the overall assessment of the model may be better judged from R_{a}^2.

The polynomial-based response surface techniques are effective in representing the global characteristics of the design space. It can filter the noise associated with design data. Since, the solution for the set of coefficients that best fits the training data is a linear least-squares problem, it is trivial compared to the solution for the NN coefficients, which is often a non-linear least-squares

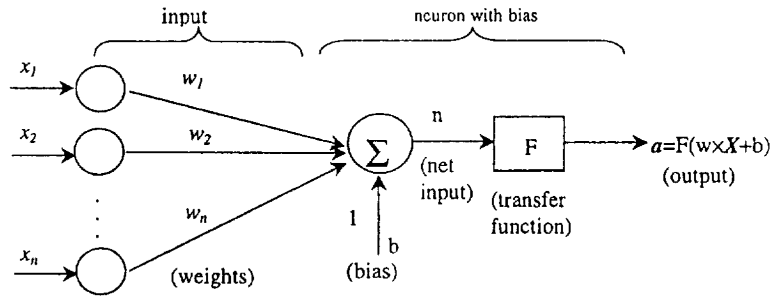


Fig. 2. Schematic of a simple neuron model.

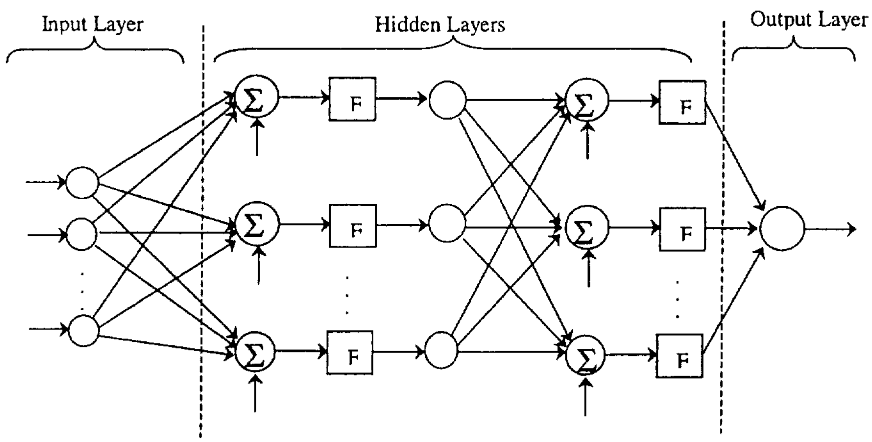


Fig. 3. Schematic of a neural network with 2-hidden layers.

problem. The linearity of the polynomial-based RSM also allows us to use statistical techniques known as *design of experiments* (DOE) to find efficient training sets. On the other hand, depending on the order of polynomial employed and the shape of the actual response surface, the RSM can introduce a substantial error in certain region of the design space. An optimization scheme requiring large amounts of data and a large evaluation time to generate meaningful results is hardly useful.

2.2. *Neural networks (NN)*

Neural networks are massively parallel computational systems comprised of simple non-linear processing elements with adjustable interconnections. Neural networks simulate human functions such as learning from experience, generalizing from previous to new data, and abstracting essential characteristics from inputs containing irrelevant data [10]. The processing ability of the network is stored in the inter-unit connection strengths or weights obtained by a process of adaptation to, or learning from, a set of training patterns. Training of a network requires repeated cycling through the data, each time adjusting the values of the weights and biases to improve performance. Each pass through the training data is called an epoch and the NN learns through the overall change in weights accumulating over many epochs. Training continues until the error target is met or until the maximum number of neurons is reached. In Fig. 2, a neuron model with multiple inputs and bias is shown.

Accordingly, the input is transmitted through a connection that multiplies it with the weight related to that connection. The bias is similar to a weight except that it has a constant input of 1. The effect of the product weight and input and the bias are added at the summing junction to form the net input for the transfer (or activation) function. In Fig. 3, a multi-layer network is shown. A layer of network includes the combination of weights, the multiplication and summing operations, the biases and the transfer functions. In a layered neural network, neurons in every layer are associated with neurons in the previous layer in such a way that the outputs of an intermediate layer are the inputs to the following layer. The layer that produces the network output is called an output layer. All other layers are known as hidden layers.

Even though research on neural network started in early 1940s, NN became quite popular around 1980s with the introduction of multi-layered NN [42] in a wide range of disciplines, including engineering. Over the last decade, NN approach has been used in the aerospace related industry. Illi et al. [17] examined the application of NN technology to an automated diagnostic and prognostic system for turbine engine maintenance. Preliminary results indicated that using NN to maintain diagnostics saves time and improves performance. Kangas et al. [18] used back-propagation NNs (BPNN) to monitor turbine engine performance and diagnose failures in real time. The application of NN technology appears to hold great promise for enhancing the effectiveness of army maintenance practices. Huang et al. [16] developed and evaluated a multi-point inverse airfoil design method using NNs. It is shown that neural network predictions are acceptable for lift and moment coefficient predictions. Time-dependent models that predict unsteady boundary layer development, separation, dynamic stall and reattachment are developed by Faller and Schereck [12] using NNs. It is demonstrated that NNs can be used to both predict and control unsteady aerodynamics effectively. Fan et al. [13] introduced a new approach for active laminar flow control that incorporates BPNN into a smart wall interactive flow control system. Convergence of the BPNN is investigated with respect to the complexity of the required function approximation, the size of the network in relation to the size of optimal solution and the degree of noise in the training data by Lawrence et al. [20]. The techniques and principles for the implementation of neural network simulators are also presented by Lawrence et al. [21]. Methods for ensuring the correctness of results avoiding duplication, automating common tasks, using assertions liberally, implementing reverse algorithms, employing multiple algorithms for the same task, and using extensive visualization are discussed. Efficiency concerns, including using appropriate granularity object-oriented programming, and pre-computing information whenever possible, are also studied. Norgaard et al. [26] used BPNN for more effective aerodynamic designs during wind tunnel testing. Four different NNs are trained to predict coefficients of lift, drag, moment of inertia, and lift drag ratio (C_L, C_D, C_M and L/D) from angle of attack and flap settings. Hybrid neural network optimization method is successfully applied to produce fast and reliable predictions of aerodynamic coefficients and to find optimal flap settings, and flap schedules. Ross et al. [30] applied BPNN to minimize the amount of data required to completely define the aerodynamic performance of a wind tunnel model. It is shown that the trained NN has a predictive accuracy equal to or better than the accuracy of the experimental measurements using only 50% of the data acquired during the wind tunnel test. BPNN is employed for rapid and efficient dynamics and control analysis of flexible systems by Sparks and Maghami [31]. It is demonstrated that NN can give very good approximations to non-linear dynamic components, and by their judicious use in simulations, allow the analyst the potential to speed up the analysis process considerably once properly trained. The high-lift performance of a multi-element airfoil is optimized by using neural-net predictions by Greenman [10].

BPNN have been successfully integrated with a gradient-based optimizer to minimize the amount of data required to completely define the design space of a three-element airfoil. It is shown that using NN reduced the amount of computational time and resources needed in high-lift rigging optimization. Greenman and Roth [14] also applied BPNN for high-lift performance of a multi-element airfoil and it is demonstrated that the trained NN predicted the aerodynamic coefficients within an acceptable accuracy defined to be the experimental error. Stepniewski and Jorgenson [32] used a singular-value decomposition-based node elimination technique and enhanced implementation of the Optimal Brain Surgeon algorithm to choose a proper NN architecture. It is demonstrated that combining both methods creates a powerful pruning scheme that can be used for tuning feed-forward connectionist models. Maghami and Sparks [23,24] also demonstrated that the methodology they developed based on statistical sampling theory guarantees that the trained networks provide a designer-specified degree of accuracy in mapping the functional relationship. The BPNN is used to fill in a design space of computational data in order to optimize flap position for maximum lift for a multi-element airfoil by Greenman and Roth [15]. A *genetic algorithm* (GA) and gradient-based optimizer are used together with NN and it is found that the demonstrated method has a higher fidelity and a reduction in CPU time when compared to an optimization procedure that excludes GA. Approximation abilities of BPNN is addressed by Lavretsky [19]. A novel matrix method for multi-input–multi-output NN is introduced and it is shown that by allowing inner layer connections as well as connections between any layers, ordered NN has superior interpolation ability when compared to conventional feed-forward NN. Stepniewski et al. [33] presented a new hybrid method that combines a bootstrap technique and a collection of stochastic optimization method such as GA for designing a NN. The method minimizes generalization error. It is demonstrated that the solutions produced by this method improve the generalization ability on the average of five to six times when compared to pruned methods.

All of the above-listed references preferred to use BPNN among the other NN choices [43–45]. This is due to the fact that BPNN strives to use a smaller number of neurons when compared to the other NNs. However, since BPNN is usually slower because at each step the error is propagated back to all the weights in the system,

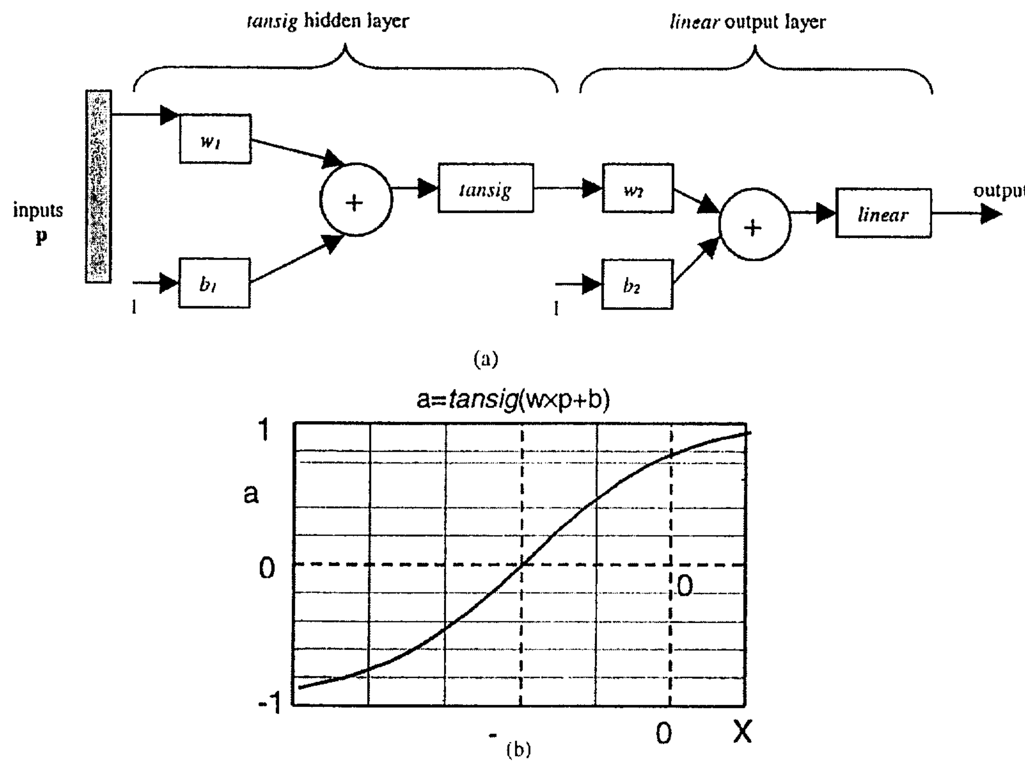


Fig. 4. (a) Back-propagation neural network architecture, (b) back-propagation transfer function, tansig.

other NNs could be more efficient than BPNN for specific problems. This article reviews the works focusing on *radial-basis* NN (RBNN) and BPNN models developed by using Matlab [43]. A comparative study for radial-basis and back-propagation approaches is also included. Brief summaries of the two approaches are given in the following sections.

2.2.1. Back-propagation neural networks (BPNN)

Back-propagation neural networks are created by generalizing the Widrow–Hoff learning rule [43,44] to multiple-layer networks and non-linear differentiable transfer functions. These networks are multi-layer networks with hidden layers of sigmoid transfer function and a linear output layer. The transfer function in the hidden layers should be differentiable and thus, either log-sigmoid or tan-sigmoid functions are commonly used. In this article, a single hidden layer with a tan-sigmoid transfer function, tansig (Fig. 4), given as $\tanh(n)$, is considered if n is the input. The maximum and minimum outputs of the function are 1 and -1, respectively.

The output of the function is given by

$$a = \text{tansig}(\mathbf{w} \times \mathbf{X} + \mathbf{b}), \tag{12}$$

where tansig is the transfer function, $\mathbf{w}$ is the weight vector, $\mathbf{X}$ is the input and $\mathbf{b}$ is the bias. For BPNN, the initial weights and biases are randomly generated and then the optimum weights and biases are evaluated through an iterative process. The weights and biases are updated by changing them in the direction of down slope with respect to the sum-squared error of the network, which is to be minimized. The sum-squared error is the sum of the squared error between the network prediction and the actual values of the output. In BPNN (Fig. 4a) the weights, $\mathbf{w}_1$, and biases, $\mathbf{b}_1$, in the hidden tansig layer are not fixed, as in the case of RBNN. Hence, the weights have a non-linear relationship in the expression between the inputs and the outputs. This results in a non-linear regression problem, which takes a longer time to solve than RBNN. Depending upon the initial weights and biases, the convergence to an optimal network design may or may not be achieved. Due to the randomness of the initial guesses, if one desires to mimic the process exactly for some purpose, it is impossible to re-train the network with the same accuracy or convergence unless the process is reinitiated exactly as before. The initial guess of the weights is a random process in Matlab. Hence to re-train the network the initial guess has to be recorded.

The number of neurons in the hidden layer of a back-propagation network is a design parameter. It should be large enough to allow the network to map the functional relationship, but not too large to cause overfitting. As a rule of thumb to choose the number of neurons in the hidden layer, Greenman [10] used $2s + 1$ where s is the summation of total number of inputs and total number of outputs and Carpenter and Barthelemy [11] used $m + 1$ where m is the number of nodes in the output layer. Once the number of neurons in the hidden layer is decided, the network design is reduced to adjusting the weighting coefficient matrices and the weighting bias vectors. These parameters for BPNN are usually adjusted using a gradient method such as the Levenberg–Marquardt technique [10,26,30,31,33]. In Matlab, BPNN can be trained by using three different training functions, trainbp, trainbpx and trainlm. The first two are based on the steepest descent method. Simple back-propagation with trainbp is usually slow since it requires small learning rates for stable learning. Trainbpx, applying momentum or adaptive learning rate, can be a considerably faster method than trainbp but trainlm, applying Levenberg–Marquardt optimization, is the most efficient since it includes improvement techniques to increase the speed and reliability of simple back-propagation networks. The Levenberg–Marquardt update rule is

$$\Delta W = (\mathbf{J}^{\mathrm{T}}\mathbf{J} + \mu\mathbf{I})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{e}, \tag{13}$$

where ΔW is the change in weight, $\mathbf{J}$ is the Jacobian matrix of the derivatives of each error with respect to each weight, i.e., $\partial e_j/\partial w_i$, $\mathbf{I}$ is the identity matrix, μ is a scalar and $\mathbf{e}$ is the error vector. If the scalar μ is large, the above expression approximates the steepest descent, while if it is small then the method reduces to the Gauss–Newton method. The Gauss–Newton method is faster and more accurate near an error minimum, so the aim is to shift towards the Gauss–Newton method as quickly as possible. Therefore, μ is decreased after each successful step and increased only when a step increases the error. The design parameters for trainlm are the number of neurons in the hidden layer, S_1, a user-defined sum square error goal, and the maximum number of epochs. The training continues until either the error goal is reached, the minimum error gradient occurs, the maximum value of μ occurs, or the maximum number of epochs has been met.

2.2.2. *Radial-basis neural networks (RBNN)*

Radial-basis neural networks are two-layer networks with a hidden layer of radial-basis transfer function and a linear output layer. The main advantage of this approach is the ability of keeping the mathematics simple and computational costs low due to linear nature of RBNN [34]. Outline of supervised learning, main application area for RBNNs and the least-squares method used together with supervised learning with linear models are explained in detail in [34]. Optimum of the regularization parameter of RBNN is also searched in this paper. A computational method for re-estimating the regularization parameter of RBNN, based on generalized cross-validation, is explained by Orr [35]. The RBNN is designed in such a way that it can adapt the width of the basis function, and it is found that it can predict better than a similar RBNN with the fixed width basis function. Orr [36] explains improvements made for to forward selection and ridge regression methods. A methodology that is a cross between regression trees and RBNN is described. The size of RBNN is also optimized based on regularization parameter in [35].

The transfer function for radial-basis neuron is radbas, which is shown in Fig. 5. radbas, given as e^{-n^2}, where n is the input, has maximum and minimum outputs of 1 and 0, respectively. The output of the function is given by

$$a = \mathrm{radbas}(\mathrm{dist}(\mathbf{w}, \mathbf{X}) \times \mathbf{b}), \tag{14}$$

where radbas is the transfer function, dist is the vector distance between the network's weight matrix, $\mathbf{w}$, and the input vector, $\mathbf{X}$ and $\mathbf{b}$ is the bias. Radial-basis transfer function radbas calculates its output according to $a = e^{-n^2}$.

In a radial basis network (Fig. 5a) each neuron in the radbas hidden layer is assigned weights, $\mathbf{w}_1$ which are equal to the values of one of the training input design points. Therefore, each neuron acts as a detector for a different input. The bias for each neuron in that layer, $\mathbf{b}_1$ is set to 0.8326/sc, where sc is the spread constant, a value defined by the user. The spread constant defines the region of influence by each neuron. The training process is then reduced to the evaluation of the weights, $\mathbf{w}_2$, and biases, $\mathbf{b}_2$, in the output linear layer, which is a linear regression problem. If the input to a neuron is identical to the weight vector, the output of that neuron is 1, since the effective input to the transfer function is zero. When a value of 0.8326 is passed through the transfer function the output is 0.5. For a vector distance equal to or less than 0.8326/$\mathbf{b}$, the output is 0.5 or more. The spread constant defines the radius of the design space over which a neuron has a response of 0.5 or more. Small values of sc can result in poor response in a domain not closely located to neuron positions; that is, for inputs that are far from the training data as compared to the defined radius, the response from the neurons will be negligible. Large values will result in low sensitivity of neurons. Since the radius of sensitivity is large, neurons whose weights are different from the input values by a large amount will still have high output thereby resulting in a flat network. The best value of the spread constant for some test data can be found by comparing σ for networks with different spread constants.

In Matlab, radial-basis networks can be designed using two different design procedures, solverbe and solverb. Both procedures require a spread constant, sc, as a design parameter; i.e., the radius of the basis in the input space to which each neuron responds. Solverbe

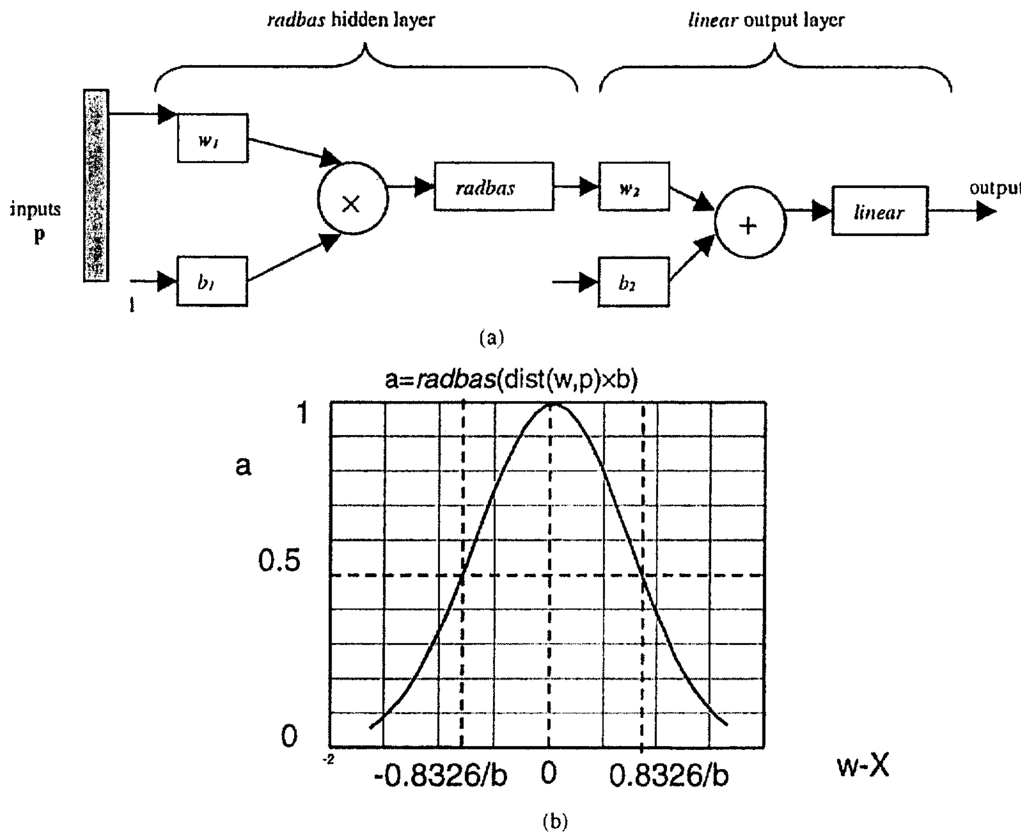


Fig. 5. (a) Radial basis neural network architecture, (b) radial-basis transfer function, radbas.

designs a network with zero error on the training vectors by creating as many radial-basis neurons as there are input vectors. Therefore, solverbe may result in a larger network than required and may fit the numerical noise. A more efficient design in terms of network size is obtained from solverb, which creates one neuron at a time to minimize the number of neurons required. At each epoch, neurons are added to the network until it satisfies a user-specified error goal. The design parameters for solverb are the spread constant, error goal, and the maximum number of epochs. The spread constant is the only network design parameter for solverbe.

Radial-basis networks may require more neurons than a comparable BPNN. However, RBNN can be designed in a fraction of the time it takes to train the standard BPNN due to non-linear regression process of back-propagation networks. Therefore, RBNN are more efficient to train when a large amount of training data is available. In [37], an effort is made to compare the accuracy and computing requirements between the radial-basis and back-propagation approaches with different sizes of training data. Vaidyanathan et al. [39] also investigated relative performances of RBNN and BPNN for gas–gas injector and supersonic turbine. As will be discussed in the following sections, among all the NN configurations, RBNN designed with solverb seems to be more consistent in performance for different data sets and RBNN, even when designed efficiently with solverb, tend to have many more neurons than a comparable BPNN with tan-sigmoid or log-sigmoid neurons in the hidden layer. The basic reason for this is the fact that the sigmoid neurons can have outputs over a large region of the input space, while radial-basis neurons only respond to relatively small regions of the input space. However, configuring an RBNN often takes less time than that required for a BPNN because the training process of RBNN is linear in nature.

2.3. *Design of experiments* (*DOE*)

In RSM, selecting the representation of the design space is a critical step because it dictates the distribution of the information available for constructing the response surface. It is well established that the predictive capability of RSM is greatly influenced by the distribution of sampling points in design space [46,47]. In order to select design points for training that minimizes the effect of noise on the fitted polynomial, design of experiment

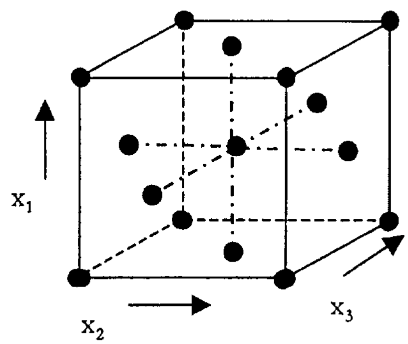


Fig. 6. Face centered composite designs (FCCD) for 3 design variables; x_1, x_2 and x_3.

(DOE) techniques can be applied. There are different types of design of experiments techniques in the literature as reported by Haftka et al. [48]. For example, Unal et al. [47] discussed the D-optimal design for the representation of the design space for a wing-body configuration of a launch vehicle. They showed that *D-optimal* design provides an efficient approach for approximating model building and multi-disciplinary optimization. Papila and Haftka [49] also applied face centered composite design (FCCD) to select the experiment points in the design space when approximating wing structural weight. Unal et al. [46,50] studied response surface modeling using orthogonal arrays (OA) in computer experiments for reusable launch vehicles and illustrated that using this technique can minimize design, development, test and evaluation cost. Unal and Dean [51] studied the robust design method based on the Taguchi method [52,53] to determine the optimum configuration of design parameters for performance, quality and cost. They demonstrated that using such a robust design method for selection of design points is a systematic and efficient approach for determining the optimum configuration. Brief summaries of FCCD, OA, and D-Optimal designs are given below.

2.3.1. Face centered cubic design (FCCD)

Face centered cubic design (FCCD) creates a design space composed of eight corners of a cube, centers of the six faces and the center of the cube. Fig. 6 shows FCCD selections for three design variables. The FCCD yields $(2^N + 2N + 1)$ points, where N is the number of design variables. It is more effective when the number of design variables is modest, say, not more than 5 or 6. The FCCD is used for fitting second-order response surface.

2.3.2. Orthogonal arrays (OA)

An orthogonal array (OA) is a fractional factorial matrix that assures a balanced comparison of levels of any factor or interaction of factors. Consider **A**, a matrix with elements of a_i^j where j denotes the row $(j = 1, 2, \ldots, n_r)$ and i denotes the column $(i = 1, 2, \ldots, n_c)$ that a_i^j belongs to, supposing that each $a_i^j \in Q = \{0, 1, \ldots, q-1\}$. **A** is called an orthogonal array of strength $t \leqslant n_c$ if in each n_r-row-by-t-column sub-matrix of all q^t possible distinct rows occur λ times [54]. Such an array is denoted by $\mathrm{OA}(n_r, n_c, q, t)$ by Owen [54].

Since the points are not necessarily at the vertices, the OA can be more robust than the FCCD in interior design space and are less likely to fail the analysis tool. Based on the design of experiment theory, orthogonal arrays can significantly reduce the number of experimental configurations.

2.3.3. D-optimal design

A D-optimal design minimizes the generalized variance of the estimates, which is equivalent to maximizing the determinant of the moment matrix, **M** [41]

$$|\mathbf{M}| = \frac{|\mathbf{X}^\mathrm{T}\mathbf{X}|}{n^{n_p}}, \tag{15}$$

where n is the number of observations and n_p is the number of terms in the model.

The D-optimal design approach makes use of the knowledge of the properties of polynomial model in selecting the design points. This criterion tends to emphasize the parameters with the highest sensitivity [48].

2.4. Optimization process

2.4.1. Search procedure

The entire optimization process can be divided into two parts: (1) RS/NN phase for establishing an approximation, and (2) optimizer phase.

In the first phase, polynomials or NN models are generated with the available training data set. In the second phase the optimizer uses the RS/NN during the search for the optimum until the final converged solution is obtained. The initial set of design variables is randomly selected from within the design space. The flowchart of the process is shown in Fig. 7.

The optimization problem at hand can be formulated as $\min\{f(x)\}$ subject to $\mathrm{lb} \leqslant \mathbf{x} \leqslant \mathrm{ub}$, where lb is the lower boundary vector and ub is the upper boundary vector of the design variables vector $\mathbf{x}$. If the goal is to maximize the objective function then $f(x)$ can be written as $-g(x)$, where $g(x)$ is the objective function. Additional linear or non-linear constraints can be incorporated if required. The optimization toolbox in *Matlab* used here employs a sequential quadratic-programming algorithm.

2.4.2. Objective function

When attempting to optimize two or more different objective functions, conflicts between them arise because of the different relationships they have with the independent parameters. An equation expressing the relationship

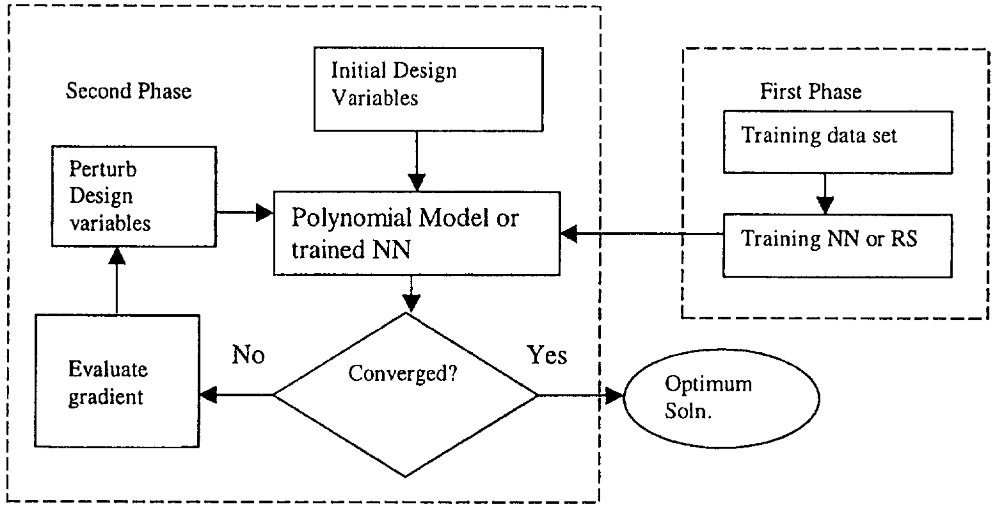


Fig. 7. The two phases of the optimization process, where Phase 1 deals with data processing/generating and Phase 2 deals with optimization.

between opposing effects of performance and weight can be employed as a criterion to guide the optimization task. Both NN and polynomial-based RS techniques can handle such multi-criteria optimization tasks in a straightforward manner by building a *composite response surface* from individual response surfaces. Such a task would have been impossible without response surface. This composite response surface is referred to as the *desirability function*. The maximization of the composite function effectively provides a compromise between the individual functions. An average of some form is normally used to represent the composite function. A geometric mean is a solution, which gives a composite function of the form

$$D = \left(\prod_{i=1}^{l} d_j \right)^{1/l}, \tag{16}$$

where D is the composite objective function, d_j's are normalized values of the objective functions and l is the number of objective functions. Each of the d_i are weighted depending upon the importance of the specific objective function. Fig. 8 shows a typical trend for a desirability function with respect to the weighting factors.

Another way of constructing a composite function is to use a weighted sum of the objective functions. The composite function can then be expressed as

$$D = \sum_{i=1}^{l} \alpha_i f_i, \tag{17}$$

where D is the composite objective function and f_i's are the non-normalized objective functions. The α_i's are dimensional parameters that control the importance of each objective function.

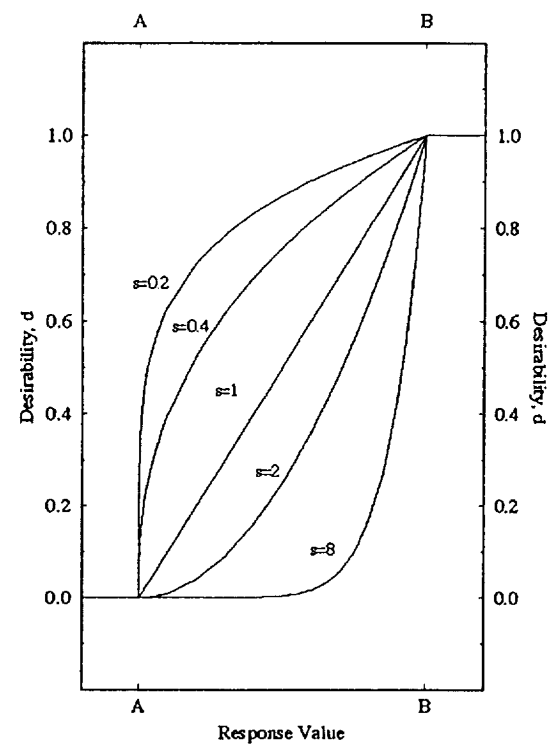


Fig. 8. Desirability function for various weight factors, s.

3. Description of the case studies

3.1. Gas–gas injector element for rocket propulsion

Development of an optimization scheme for injector design called methodology for optimizing the design of

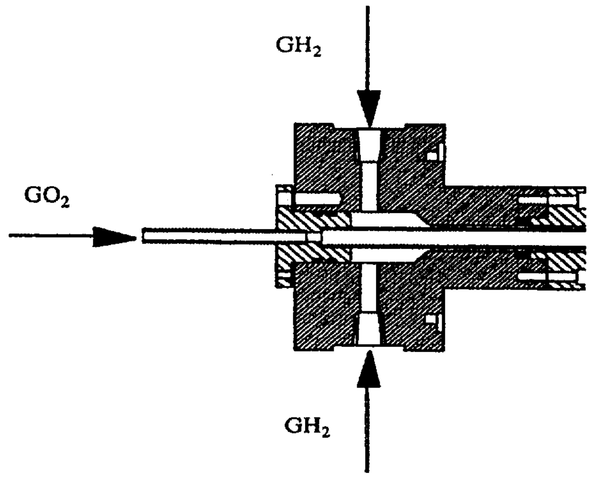


Fig. 9. Schematic of shear co-axial injector element.

injectors (*method i*) has been reported by Tucker et al. [55,56]. Method *i* is used to generate appropriate injector design data and then guide the designer toward an optimum design subject to his specified constraints. As reported, method *i* uses the polynomial-based RSM to facilitate the optimization. The RSM approach is to conduct a series of well-chosen experiments (empirical, numerical, physical or some combination of the three) and use the resulting information to construct a global approximation (response surface) of the measured quantity (response) over the design space. A standard constrained optimization algorithm is then used to interrogate the response surface for an optimum design. Neural network was also used in the design of shear co-axial injector element by Shyy et al. [38], and Tucker et al. [55,56] along with the polynomial-based RSM. Three different injector types are considered, namely, shear co-axial injector element, an impinging injector element, and swirl co-axial injector element.

3.1.1. Shear co-axial injector element

The initial demonstration of method *i* by Tucker et al. [55] focused on a simple optimization of a shear co-axial injector element (Fig. 9) with gaseous oxygen (GO_2) and gaseous hydrogen (GH_2) propellants. The goal is to maximize the energy release efficiency, ERE while minimizing the chamber wall heat flux, Q. This is achieved by maximizing a composite objective function given by

$$D = (d_{\text{ERE}} d_Q)^{1/2}, \tag{18}$$

where the normalized functions are defined as in Eqs. (19) and (20). In the case where a response should be maximized, such as ERE, the normalized function takes the form

$$d_{\text{ERE}} = \left(\frac{\text{ERE} - A}{B - A}\right)^s \quad \text{for } A \leqslant \text{ERE} \leqslant B, \tag{19}$$

where B is the target value and A is the lowest acceptable value. Here, d_{ERE} is set to 1 for any $\text{ERE} > B$ and $d_{\text{ERE}} = 0$ for $\text{ERE} < A$. The choice of s is made based on the subjective importance of this objective in the composite desirability function. In the case where a response is to be minimized, such as Q, the normalized function takes the form

$$d_Q = \left(\frac{E - Q}{E - C}\right)^t \quad \text{for } C \leqslant Q \leqslant E, \tag{20}$$

where C is the target value and E is the highest acceptable value. Here, d_Q is set to 1 for any $Q < C$ and $d_Q = 0$ for $Q > E$. A, B, C, and E are chosen according to the designer's priorities or, as in the present article, simply as the boundary values of the domain of ERE and Q. The value of t is again chosen to reflect the importance of the objectives in the design. In the study carried out, A and B are equal to 95.0 and 99.9, respectively. The values of C and E are equal to 0.48 and 1.1, respectively.

The design data was generated using an empirical design methodology developed by Calhoon et al. [57]. These researchers conducted a large number of cold-flow and hot-fire tests over a range of propellant mixture ratios, propellant velocity ratios and chamber pressure for shear co-axial, swirl co-axial, impinging, and premixed elements. The data were correlated directly with injector/chamber design parameters, which are recognized from both theoretical and empirical standpoints as the controlling variables. For the shear co-axial element, performance, as measured by energy release efficiency, ERE, is obtained using correlations taking into account combustor length, L_{comb} (length from injector to throat) and the propellant velocity ratio, V_f/V_o. The nominal chamber wall heat flux at a point just downstream of the injector, Q_{nom}, is calculated using a modified Bartz equation and is correlated with propellant mixture ratio, O/F, and propellant velocity ratio, V_f/V_o to yield the actual chamber wall heat flux, Q. The objective in the initial demonstration of method *i* was to maximize injector performance while minimizing chamber wall heat flux (lower heat fluxes reduce cooling requirements and increase chamber life) and chamber length (shorter chambers lower engine weight). The data used to generate the polynomials and train the network are given in Tables 36–38. The quality of the response surface and neural networks are evaluated using 20 additional design points different from those used to generate the models (Table 39).

3.1.2. Impinging injector element

The empirical design methodology of Calhoon et al. [57] uses the oxidizer pressure drop, ΔP_o, fuel pressure drop, ΔP_f, combustor length, L_{comb}, and the impingement half-angle, α as independent variables. For this injector design, the pressure drop range is set to 10–20%

of the chamber pressure due to stability considerations. The combustor length, defined as the distance from the injector to the end of the barrel portion of the chamber, ranges between 2 and 8 inches. The impingement half angle is allowed to vary from 15 to 50°. Dependent variables include ERE (a measure of element performance), wall heat flux, Q_w, injector heat flux, Q_{inj}, relative combustor weight, W_{rel}, and relative injector cost, C_{rel}.

The conditions selected for this example are:

$$P_c = 1000\,\text{psi},$$

$$\text{MR} = 6,$$

$$m_{GO_2} = 0.25\,\text{lb}_m/\text{s},$$

$$m_{GH_2} = 0.042\,\text{lb}_m\text{s}. \tag{21}$$

The gaseous propellants are injected at a temperature of 540R. As noted above, the empirical design methodology used to characterize the ERE and Q_w was developed by Calhoon et al. [57]. This methodology uses a quantity called the normalized injection momentum ratio to correlate the mixing at the different design points for the triplet element. They define this quantity as

$$\text{MR}_{ni} = \frac{2.3 m_0 u_0}{m_f u_f \sin\alpha}. \tag{22}$$

The maximum mixing, and thus maximum ERE, occurs at an MR_{ni} of 2.0. Since the propellant mass flow rates are fixed, only the propellant velocities and the impingement half-angle influence the normalized injection momentum ratio. The velocities are proportional to the square root of the respective pressure drops across the injector, ΔP_o and ΔP_f. For the flow conditions and variable ranges considered in this problem, MR_{ni} ranges from 3.2 to 17.8. Accordingly, lowering ΔP_o, raising ΔP_f, increasing α, or some combination of these actions will increase ERE. The wall heat flux is correlated with the propellant momentum ratio as defined by

$$\text{MR} = \frac{m_0 u_0}{m_f u_f}. \tag{23}$$

For the F–O–F triplet element, i.e. the impingement injector element, the maximum wall heat flux occurs at a momentum ratio of approximately 0.4. High heat flux is the result of over-penetration of the fuel jet, which produces a high O/F in the wall region. For the flow conditions and variable ranges considered in this effort, MR ranges from 1.06 to 2.11. Hence, increasing the value of this ratio by either increasing ΔP_o or decreasing ΔP_f lowers the wall heat flux.

The heat flux seen by the injector face, Q_{inj}, is qualitatively modeled by the impingement height, $H_{impinge}$. Here the notion is that, as the impingement height decreases, the combustion occurs closer to the injector face, causing a proportional increase in Q_{inj}. Thus, for the purposes of

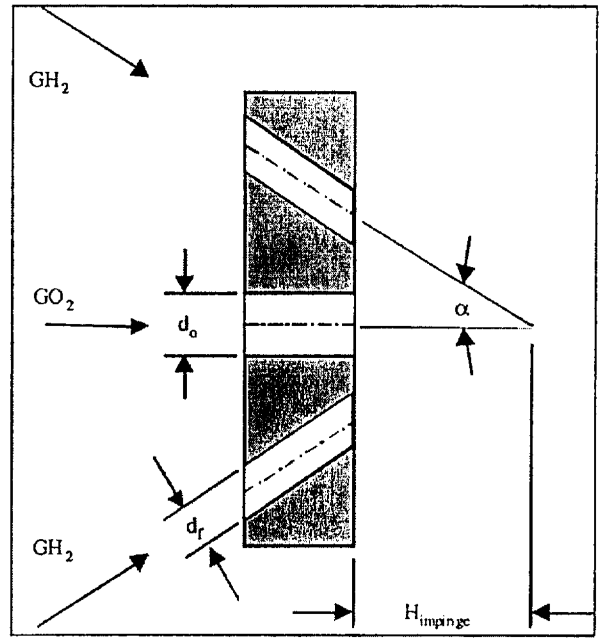


Fig. 10. Schematic of impingement injector element.

this exercise, Q_{inj} is modeled as the reciprocal of the $H_{impinge}$. Impingement height is a function of α and ΔP_f. Fig. 10 shows that as α is increased, $H_{impinge}$ is shortened. The dependence of $H_{impinge}$ on the fuel orifice diameter, d_f, and thus, ΔP_f, results from making the freestream length of the fuel jet, L_{fs}, a function of d_f. For each ΔP_f, L_{fs} was set to six times d_f for an impingement half-angle of 30°. So, as d_f increases (corresponding to decreasing ΔP_f), L_{fs} increases, as does $H_{impinge}$.

The models for W_{rel} and C_{rel} are simple but represent the correct trends. W_{rel} is a function only of L_{comb}, the combustor length from injector face to the end of the chamber barrel section. The dimensions of the rest of the thrust chamber assembly are fixed. So, as L_{comb} increases, W_{rel} increases accordingly. The model for C_{rel} is based on the notion that smaller orifices are more expensive to machine. Therefore, C_{rel} is a function of both propellant pressure drops. As the ΔP increases, the propellant velocity through the injector increases and the orifice area decreases. So, as either, or both, ΔP_o and ΔP_f increase, C_{rel} increases.

The system variables given above and independent variables (constrained to the previously noted ranges) are used to generate the design data for element optimization studies. Since propellant momentum ratio is an important variable in the empirical design methodology, a matrix of momentum ratios was developed over the 100–200 psi propellant pressure drop range. The matrix of 49 combinations of fuel and oxidizer pressure drops is shown in Table 40 where momentum ratios range from 1.06 to 2.11. Nine pressure drop combinations, eight

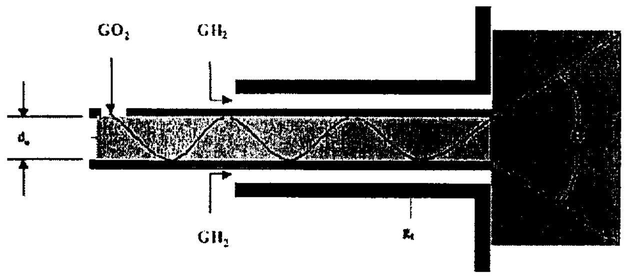


Fig. 11. Swirl co-axial injector element schematic.

around the border and one in the middle, were selected for use in populating the design database. These nine points are highlighted in Table 40 in bold type.

Detailed design results for the case with both ΔP_o and ΔP_f at 200 psi are shown in Table 41. Similar data was generated for the other eight pressure drop combinations. There are 20 combinations of L_{comb} and α for each ΔP combination, making a total of 180 design points selected. Seventeen of these were outside the database embodied by the empirical design methodology, resulting in 163 design points actually being evaluated. The data trends are as expected. ERE, for a given ΔP combination, increases with increasing L_{comb} and α. The increased L_{comb} provides more residence time for the propellants to mix and burn. Increasing α increases the radial component of the injected fuel, thus providing better mixing. The wall heat flux is constant for a given ΔP combination. Impingement height increases with increasing α. Relative combustor cost increases with increasing L_{comb} and the relative injector cost is constant for a given ΔP combination.

3.1.3. Swirl co-axial injector element

The chamber pressure, mixture ratio, and propellant flow rates selected for this example are:

$$P_c = 1000\,\text{psi},$$

$$\text{MR} = 6,$$

$$m_{GO_2} = 0.25\,\text{lb}_m/\text{s},$$

$$m_{GH_2} = 0.042\,\text{lb}_m\text{s}. \tag{24}$$

The gaseous propellants are injected at a temperature of 540*R*. Fig. 11 shows that the GO_2, flowing in the center post of the element, exits the element with both radial and axial velocity components. This effect is achieved by introducing the GO_2 tangentially into the center post through small slots. When the GO_2, under hydrostatic head, is forced through the tangential slots, part of the pressure head is converted into a velocity head, causing a rotational velocity in the element. For a specified ΔP_o and swirl angle, Θ, the number and size of tangential slots, the discharge coefficient, the GO_2 center post diameter, d_o, and the radial and axial GO_2 velocity components, V_{or} and V_{oa} are calculated. These quantities are then used to determine the dependent variables for each design condition.

The element ERE, calculated according to the empirical design methodology of Calhoon et al. [57], is a function of all four independent variables noted above. A cold flow mixing efficiency, $E_{m,90}$, for $\Theta = 90°$, is correlated by

$$E_{m,90} = 100 - 5\ln\left[\frac{K_s}{L_{cold}/d_o}\right]. \tag{25}$$

The cold flow mixing length, L_{cold}, is correlated from a known chamber length, L_{comb}. The GO_2 post diameter, d_o, is a function of ΔP_o and Θ. Smaller values of d_o correspond to large values of ΔP_o and smaller swirl angles. The empirical swirl factor, K_s, is a function of the normalized differential injection velocity, $(V_f - V_o)/V_o$. K_s increases with increasing normalized differential injection velocity for the range of propellant velocities considered in this effort. For fixed propellant mass flow rates, the velocities V_o and V_f are functions of their pressure drops across the injector, ΔP_o and ΔP_f, respectively. For a given ΔP_o, V_o also depends on the swirl angle. Lower V_o's are a product of higher swirl angles. Cold flow mixing is thereby enhanced with higher values of V_o (i.e. ΔP_o) and L_{comb}. Lower values of V_f (i.e. ΔP_f) and Θ also tend to enhance cold flow mixing.

A fractional factor, f_s, is applied to $E_{m,90}$ to account for the lower levels of cold flow mixing found with swirl angles less than 90°. The resultant measure of cold flow mixing, $E_{m,\Theta}$, is a product of $E_{m,90}$ and f_s. This factor, for a given design, is a function of the normalized differential injection velocity and the ratio of radial to axial GO_2 velocity, V_{or}/V_{oa}. Increasing values of both quantities increase f_s, with a value of $f_s = 1$ being found at $V_{or}/V_{oa} = 1$ ($\Theta = 90°$) for all values of $(V_f - V_o)/V_o$. Larger values of f_s increase cold flow mixing. These values are found at low ΔP_o and high ΔP_f and Θ. There is no dependency of f_s on chamber length. These trends are opposite those noted above. Finally, ERE is proportional to $E_{m,\Theta}$.

The wall heat flux curve from the Calhoon et al. [57] methodology is fairly flat, varying only about 10% from high to low for the range of pressure drops considered in this effort. Q_w decreases with increasing V_o (high ΔP_o and low Θ) and decreasing V_f (low ΔP_f). That Q_w would decrease with increasing V_o is counter to intuition. It seems that high values of V_o, for any Θ, would result in higher mixture ratios in the wall region, as is the case for liquid O_2. Calhoon et al. [57] do not discuss this effect.

The heat flux seen by the injector, Q_{inj}, is actually modeled by the distance from the injector at which the propellant streams intersect. This axial distance is measured at the radial position corresponding to the center of the co-axial fuel annulus, or gap. It is here that the streams begin to mix and burn. This measure is qualitative, but captures the trend for injector design. The axial distance is affected directly by the swirl angle, and indirectly by the propellant pressure drops. Q_{inj} decreases with decreasing swirl angle, increasing GO_2 pressure drop and decreasing GH_2 pressure drop. Swirl angle has the largest effect, while ΔP_o is the least significant factor.

The relative combustor weight, W_{rel}, is simply a function of the combustor length, L_{comb}, the distance from the injector to the end of the barrel portion of the chamber. The longer the combustor, the more it weighs.

The relative injector cost, C_{rel}, is a function of the fuel gap width and the width of the tangential slots used to induce the swirl in the GO_2 center post. Larger values of both variables result in lower machining costs, and thus lead to lower injector cost. The fuel gap width increases with increasing ΔP_o, and decreasing values of ΔP_f and Θ. Swirl slot width increases with lower values of ΔP_o and Θ. Overall, C_{rel} decreases with increasing ΔP_o and decreasing ΔP_f and Θ. Fuel pressure drop and swirl angle are the most significant factors.

A matrix of propellant pressure drop combinations was developed and nine combinations were selected for use in populating the design database. There are 20 combinations of L_{comb} and Θ for each ΔP combination, making a total of 180 design points selected.

In the work by Tucker et al. [55,56], method *i* uses the response surface method (RSM) to find optimal values of ERE, Q_w, Q_{inj}, W_{rel} and C_{rel} for acceptable values of ΔP_o, ΔP_f, L_{comb} and Θ. Five full quadratic response surfaces are constructed by using JMP.

In the current case, it is desirable to maximize ERE and while simultaneously minimizing Q_w, Q_{inj}, W_{rel} and C_{rel}.

3.2. *Supersonic turbine for reusable launch vehicles*

Supersonic turbines that drive fuel or oxidizer turbopumps in rocket engines are of great interest to the next generation space propulsion industry, including the *reusable launch vehicles* (RLV). They are complex, high-speed devices that produce shaft power by ducting the flow of hot gasses over specially shaped blades on a wheel. For rocket engine applications, maximizing the vehicle payload for a given turbine operating condition is the ultimate goal. The flow path should be designed in such a way that it wastes less energy so that turbine temperatures or the mass flow rate can be reduced, or the turbine can be made smaller, increasing the efficiency (or specific impulse) of the rocket engine. Any gain in turbine efficiency will be reflected in reduced propellant consumption, resulting in an increase in the payload. However, higher turbine performance usually entails multistage designs, which are heavier. The design of a supersonic turbine often involves a considerable number of design variables with structural and aerodynamic constraints. With the number of design parameters involved, the overall procedure of design optimization of supersonic turbines becomes a challenging task.

Papila et al. [58] have conducted a global optimization investigation to perform the preliminary design of the supersonic turbines, including the selection of the number of stages and design variables. From one- to two- to three-stage turbines, the number of design variables increases substantially. In shape design, from vane to blade, from stage to stage, and from 2-D to 3-D, not only does the number of design variables increase, but also the interactions among design variables become more complicated. Papila et al. [58] intended to investigate the individual, as well as collective effects of design variables by varying the design scope systematically. Vaidyanathan et al. [39] have used the data of the one-stage turbine to conduct a comparative study between RSM and NN.

For the preliminary design stage, single-, two- and three-stage turbines are considered. The design variables can be separated into two categories, one related to geometry and the other to performance. They are summarized as follows:

(1) *Geometric inputs*: The geometric inputs are needed to layout the turbine meridional geometry, e.g., mean diameter, last rotor annulus area, blade height ratio between the first vane and the last rotor blade (linear distribution of blade heights is assumed between the first vane and the last rotor blade), vane and blade axial chords.

(2) *Performance inputs*: The performance inputs are needed to calculate the turbine efficiency, e.g., speed (RPM), number of stages, blade row reaction, and work split (if more than 1 stage is investigated).

For single-stage turbine, six design parameters (Table 3) are selected. These are (1) the mean diameter, (2) speed (RPM), (3) exit blade annulus area, (4) vane axial chord, (5) blade axial chord, (6) stage reaction.

For two-stage turbine, there are 11 design parameters (Table 3), namely, (1) the mean diameter, (2) RPM, (3) exit blade annulus area, (4) first blade height (% of exit blade), (5) first vane axial chord, (6) first blade axial chord, (7) second vane axial chord, (8) second blade axial chord, (9) first stage reaction, (10) second stage reaction, and (11) first stage work fraction. Note that second stage work fraction is not a design parameter since it can be calculated by using first stage work fractions, i.e., $w_{f2} = 1 - w_{f1}$.

There are 15 (Table 3) design parameters for three-stage turbine. These are (1) mean diameter, (2) speed (RPM), (3) exit blade annulus area, (4) first blade height (% of exit blade), (5) first vane axial chord, (6) first blade axial chord, (7) second vane axial chord, (8) second blade

Table 3
Design variables and design space for single-, two- and three-stage turbines (all geometric design variables are normalized by the baseline values)

Single-stage			Two-stage			Three-stage		
Variable	Lower limit	Upper limit	Variable	Lower limit	Upper limit	Variable	Lower limit	Upper limit
Mean diameter, D	0.50	1.50	Mean diameter, D	0.50	1.50	Mean diameter, D	0.50	1.50
Speed, RPM	0.70	1.30	Speed, RPM	0.70	1.30	Speed, RPM	0.70	1.30
Blade annulus area, A_{ann}	0.70	1.30	Blade annulus area, A_{ann}	0.70	1.30	Blade annulus area, A_{ann}	0.70	1.30
Vane axial chord, c_v	0.39	1.71	1st Blade height (% of exit blade), h_1	0.90	1.50	1st Blade height (% of exit blade), h_1	0.90	1.50
Blade axial chord, c_b	0.26	1.14	1st Vane axial chord, c_{v1}	0.39	1.71	1st Vane axial chord, c_{v1}	0.39	1.71
Stage reaction, sr	0.0%	50%	1st Blade axial chord, c_{b1}	0.26	1.14	1st Blade axial chord, c_{b1}	0.26	1.14
			2nd Vane axial chord, c_{v2}	0.21	1.41	2nd Vane axial chord, c_{v2}	0.21	1.41
			2nd Blade axial chord, c_{b2}	0.17	1.13	2nd Blade axial chord, c_{b2}	0.17	1.13
			1st Stage reaction, sr_1	0.0%	50%	3rd Vane axial chord, c_{v3}	0.21	1.41
			2nd Stage reaction, sr_2	0.0%	50%	3rd Blade axial chord, c_{b3}	0.17	1.13
			1st Work fraction, wf_1	50%	85%	1st Stage reaction, sr_1	0.0%	50%
						2nd Stage reaction, sr_2	0.0%	50%
						3rd Stage reaction, sr_3	0.0%	50%
						1st Work fraction, wf_1	40%	80%
						2nd Work fraction, wf_2	30%	10%

axial chord, (9) third vane axial chord, (10) third blade axial chord, (11) first-stage reaction, (12) second-stage reaction, (13) third-stage reaction, (14) first-stage work fraction, (15) second-stage work fraction. Note that third-stage work fraction is not a design parameter since it can be calculated by using first- and second-stage work fractions, i.e., $w_{f3} = 1 - (w_{f1} + w_{f2})$.

The composite objective function chosen by Papila et al. [58] for design optimization corresponds to the payload increment, Δpay, versus turbopump efficiency and weight. The relation between Δpay and these two parameters can be developed as follows based on mission profile studies, engine balance perturbation and some detailed turbopump layout and stress information gained from other proprietary programs:

$$\Delta\text{pay} = c_1 \times (\eta - \eta_b) \times 100 - (W - W_b), \tag{26}$$

where η_b is the baseline efficiency and W_b is the baseline weight. The constant c_1 indicates that for every point in efficiency gained, the amount of payload capacity of the RLV is increased c_1 per turbopump. Therefore, Δpay function represents the amount of increase in payload capacity. The results of both payload increment based and composite desirability function-based optimization are illustrated for one-, two-, and three-stage designs. The results of both payload increment based- and composite desirability function-based optimization are illustrated for one-, two-, and three-stage designs in the following chapters.

Two structural constraints are considered by Papila et al. [58]. In axial turbines the product of the blade exit annulus area and the RPM square, i.e., AN^2 is an indication of the blade centrifugal stress, which should bound the speed of the turbine. In addition, the disk stresses are also a restriction. In the turbomachinery industry, the maximum stress value due to disk burst is often represented by a pitchline velocity limit, i.e., V_{pitch}. The pitchline velocity can be calculated by multiplying RPM and the mean radius.

3.3. Turbulent planar diffuser

The goal was to accomplish maximum pressure recovery by optimizing the wall contours. The flow is incompressible and fully turbulent with a Reynolds number of 10^5, based on the inlet throat half-width, D. The overall geometry is defined by the ratio of inlet and outlet areas, and the diffuser length-to-height ratio. In this study the length-to-height ratio is fixed at 3.0, and the area ratio at 2.0. The shape of the diffuser wall is designed for optimum performance, with five design variables represented by B-splines. The CFD model is based on the full Reynolds-averaged Navier–Stokes equations, with the k–ε two-equation turbulence model in closure form. At the inlet of the flow domain, a uniform flow distribution is specified. Detailed discussion of this study can be found in [59].

3.3.1. Objective

The dimensionless pressure recovery coefficient C_p is introduced as the objective function to be maximized:

$$F = C_p = \frac{\Delta p}{1/2\,\rho u_{\text{inl}}^2}. \tag{27}$$

Here Δp is the static pressure difference between channel cross sections up- and downstream of the diffuser, respectively, ρ is the fluid density, and u_{inl} is the inlet mean velocity. Inlet and outlet static pressures are averaged, even though the pressure distribution is nearly uniform due to well-developed flow at the considered cross sections. The CFD model uses a symmetry condition along the channel center axis, and has a computational mesh consisting of 120×50 cells including a long outlet section to establish a fully developed exit profile. The overall geometry of the two-dimensional planar diffuser, see Fig. 12, is defined by the ratio of inlet and outlet areas, AR, and the diffuser length/height ratio, $L = D$, where L is the axial length of the diffuser. In this study the ratio of L to D is fixed at 3.0, and the area ratio AR at 2.0. Expressed in terms of the inlet half-width D, the horizontal position of the inlet is $1D$, while the horizontal position of the outlet is $10D$. The shape of the diffuser wall is designed for optimum performance, and to this end two separate cases of wall parameterizations were evaluated by Madsen et al.: (1) a two design variable case, where a polynomial describes wall shapes, and (2) a five design variable case that uses B-splines. Even though two different curve descriptions are used in the two cases, the most noteworthy difference seen from the point of view of the RSM lies in the problem size.

3.3.2. Geometric representation

For shape parameterization in more variables, B-splines were preferred to natural splines (piecewise polynomials), although the latter technique is closer to the polynomial representation. B-splines excel in the predictable way that control points influence curve shape, and in the *local control*, which prevents small changes in a control point position from propagating over the entire curve. Combined with low computational cost, these advantages have contributed to B-spline curves becoming a standard geometric modeling technique in computer-aided design.

A B-spline is given in a parametric form as $\mathbf{p}(u)$:

$$\mathbf{p}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(u). \tag{28}$$

A set of blending functions $N_{i;k}$ combines the influence of $n+1$ control points $\mathbf{P}_i$, over the range of the parametric variable u. The blending functions $N_{i;k}$ are recursively

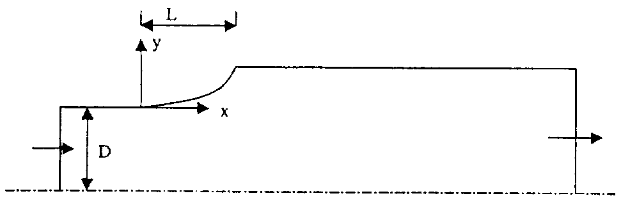


Fig. 12. Two-dimensional symmetric diffuser subjected to shape optimization in terms of pressure recovery measured between in- and outlet.

determined polynomials with degree $k-1$, where the parameter k dictates the order of continuity of the curve, and thus how many control points influence a curve segment. In this work k is 8, which corresponds to C^6-continuity. The number of control points is 8 as well—two endpoints, five design variables and one point used for prescribing the inlet slope.

B-splines have an approximating nature, in that they do not necessarily pass through control points, except for fixed curve endpoints. The slope at a curve endpoint is tangential to a straight line connecting the endpoint and the first control point, and may be prescribed by placing an additional fixed control point near the endpoint.

Experimental and numerical evidence indicates that maximum pressure recovery in diffusers occurs at the border of appreciable flow separation. For this reason, strongly separated diffuser flows should be avoided, making it reasonable to restrict the design space to monotonic wall shapes. While the approximation accuracy does of course benefit from the reasonable design space approach, it is equally important in the present example that monotonicity constraints eliminate convergence problems associated with CFD-analysis of odd, non-monotonic designs.

The parametric form in which B-splines are defined makes it non-trivial to derive monotonicity constraints analytically, so instead a constraint approximation $\hat{G}$ was set up in the form of a response surface for the minimum wall slope G. Then, observing the inequality constraint $G \geqslant 0$ implies a positive wall slope and thus monotonicity throughout. Since B-splines are inexpensive to generate, 9^5 (59049) B-splines were computed (requiring only seconds to generate) and used for fitting a quadratic response surface. The approximation to the monotonicity constraint precludes some designs that satisfy the exact monotonicity requirement. However, the effect of these inaccuracies on the solution of the optimum design problem is negligible.

The regression analysis, to find 21 polynomial coefficients in five dimensions, is based on a 35-point D-optimal design. The surplus of analyses is generally required for reducing the sensitivity to numerical noise and to errors due to the simplified representation as a quadratic polynomial. Again, a pool of candidate points was created, this time using nine levels for each variable (values ranging from 0.0 to 1.0), and then checking the monotonicity of the B-splines for each of the $9^5 = 59\,049$ designs. It should be noted that limiting the y-coordinate of the control points to a variation in the range [0:0; 1:0] is a somewhat artificial requirement, as monotonic shapes exist with coordinates slightly outside this range. A total of 20 864 points are monotonic in wall shape. This relatively large percentage of acceptable cases reflects the smoother nature of approximating curves. Had a non-segmented polynomial curve representation been used, the condition of monotonicity in the control points would alone have reduced the number of feasible design points to less than 1% of those inside a five dimensional box. As in the two-design-variable case, the subset of D-optimal points was found using the JMP.

3.4. Low Reynolds number wing model

3.4.1. Training data

The aerodynamic model, a rectangular wing with a NACA 5405 airfoil cross-section (Fig. 13) is designed for low Reynolds number (Re = 10^4–10^6) flows. Since airfoil performance decreases at low Reynolds number flights, attempts to shrink the overall aircraft size while trying to keep sufficient lifting areas result in low aspect ratio wing planforms. As aspect ratio decreases, the percentage of the wing area affected by the tip vortices increases, creating a 3-D flow field over most of the field. Therefore, the analysis of such flows should consider the effects on performance and the effects of both the airfoil geometry (such as maximum camber) and the wing

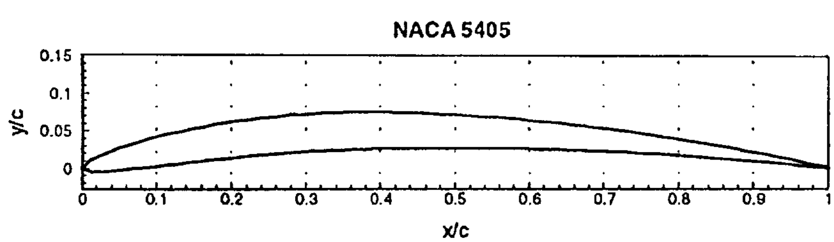


Fig. 13. NACA 5405 profile.

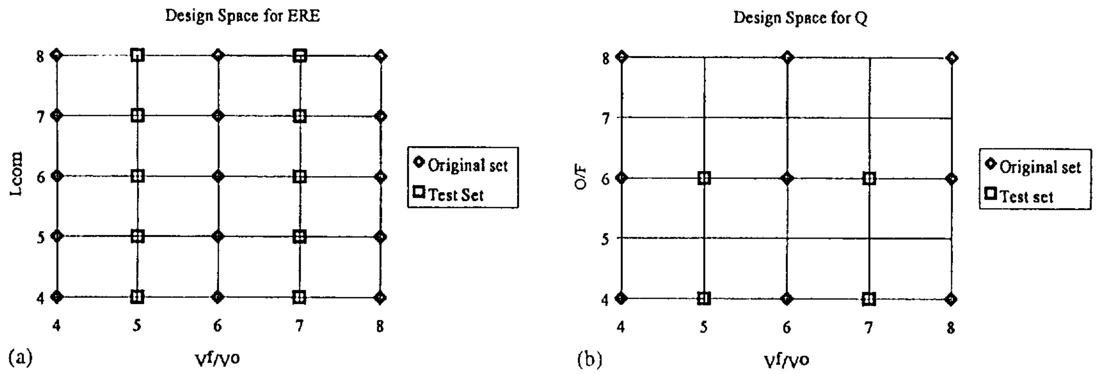


Fig. 14. Design space for the shear co-axial injector: (a) ERE (15 original points, 10 test points), (b) Q (9 original points, 4 test points).

geometry (such as aspect ratio). In this study, the aerodynamic analysis is based on a 3-D potential flow solver, PMARC, and a 2-D coupled inviscid-viscous flow solver, XFOIL$^-$. The lift coefficients, C_L, and drag coefficients, C_D, for various maximum camber, y_c, aspect ratios, AR, and angles-of-attack, α, at fixed Reynolds number, $\mathrm{Re} = 2.0 \times 10^5$, and thickness ratio, $y_t = 5\%$, are used to correlate the aerodynamic performance, measured by the power index, $C_L^{3/2}/C_D$, which appears explicitly in steady flight required-power equation. Aspect ratio and maximum camber form the input vector, **p** and $C_L^{3/2}/C_D$ forms the target output vector, **a**, as shown below:

$$\mathbf{p} = \begin{bmatrix} \mathrm{AR} \\ y_c \end{bmatrix}_{2\times\mathbf{R}} \quad \mathbf{a} = [C_L^{3/2}/C_D]_{1\times\mathbf{R}}, \tag{29}$$

where **R** is the number of input vectors of the training data.

For the 3-D wing case, the maximum camber varies between 0.0 and 0.1 and the aspect ratio varies between one and five. Three different training data sets are used out of the available data as shown in Table 34. Table 35 summarizes the test data sets used for prediction for this case. A *simulation*, referred in these tables, consists of two input variables: AR and y_c and the output variable: $C_L^{3/2}/C_D$.

4. Assessment of data processing and optimization capabilities

Of all the cases considered in this article, the impingement injector element, swirl co-axial injector element, two-stage supersonic turbine and turbulent flow diffuser help elucidate the effectiveness of using polynomial-based RSM. The shear co-axial injector element, one-stage turbine and two-dimensional wing model are used to carry out a comparative study between RSM and NN. The size of the data sets used in these studies varies from very modest to large (from 9 to 2235 data points).

In the following, we synthesize the studies of Papila et al. [37,58], Madsen et al. [59], Shyy et al. [38], Tucker et al. [55,56] and Vaidyanathan et al. [39]. We first review the data processing capabilities then evaluate the performance of the optimization techniques. For both NN and polynomials, one needs to first decide the most appropriate constructions for a given data set. For the NN, the choices are usually (1) the number of neurons, and (2) the error goals. Furthermore, the spread constant (for RBNN) and the number of hidden layers (for BPNN) can be specified. In this article, the BPNN and RBNN will be limited to the two-layer form.

4.1. Shear co-axial injector

4.1.1. Polynomial fits

According to the injector model developed by Calhoon *et al.* [57], injector performance, as measured by ERE depends only on the velocity ratio, V_f/V_o, and combustion chamber length, L_{comb}. Examination of the original data set in Tables 36–39 indicates 15 distinct design points for ERE. Since chamber wall heat flux is dependent only on velocity ratio, V_f/V_o, and oxidizer to fuel ratio, O/F, there are nine distinct design points for Q. The design space for this effort is depicted in Fig. 14. For ERE, the five distinct chamber lengths offer the potential for a fourth-order polynomial fit in L_{comb}, while the three different velocity ratios limit the fit in V_f/V_o to second order. Quadratic and cubic response surfaces for both ERE and Q have been generated for evaluation. These response surfaces represent reduced models accomplished by term elimination from the full surface using t-statistics as described earlier. The above-noted limitations on the data limit the cubic surfaces to be third order in L_{comb} only.

Based on the adjusted rms-error, Vaidyanathan et al. [39] have concluded that the cubic fit is more accurate

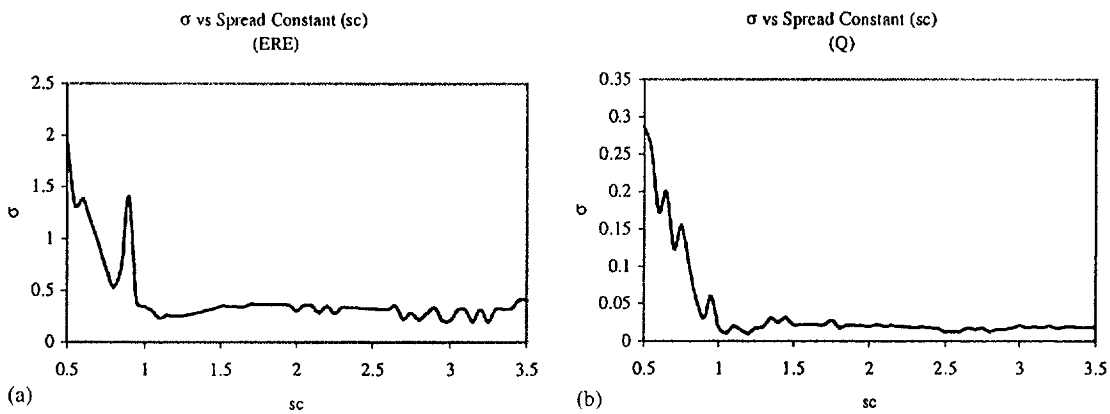


Fig. 15. Comparison of σ for different NN designed with solverbe for shear co-axial gas–gas injector: (a) ERE (%) and (b) Q (Btu/in^2 s).

than the quadratic fit for ERE. The adjusted rms-error for the quadratic and cubic response surfaces of ERE are 0.211 and 0.083, respectively. The cubic fit, by this measure, is superior for ERE. However, the error is almost identical in the case of Q for both the quadratic (0.039) and cubic (0.040) surfaces, apparently due to the very small number of points available for the curve fit. The additional terms in the cubic fit relative to the quadratic fit do not improve the mapping of the response surface for Q.

4.1.2. Construction of RBNN

In the case of the injector design there are two objectives, namely ERE and Q. Fig. 15 gives the variation of σ for the network designed with solverbe for these objective functions. In case of solverb the error goal during training also defines the accuracy of the network. An objective of fitting a numerical model is to remove the noise associated with the data. A model, which maps exactly as solverbe does, will not eliminate the noise, whereas solverb will. Fig. 15 shows that for low values of spread constant, the NN has a poor performance. As the spread constant increases σ asymptotically decreases. However, as demonstrated by Fig. 15a the performance of the network can deteriorate for higher values of the spread constant. The region with a large variation in σ is highly unreliable because this indicates a high sensitivity of the model to a small variation of spread constant and possibly the test data, in this region. Hence the desirable spread constant is selected from the region where the performance of the network is relatively consistent.

Fig. 16 gives the variation of σ for the network designed with solverb for the objective functions of ERE and Q. It also shows the influence of error goal on the network. Generally if a network maps the training data accurately it can be expected to perform efficiently with the test data. However, accurately mapping noisy data may result in poor prediction capabilities for the network. The variation in the performance is not significant except for the ERE and Q network (Fig. 16), where the poor performance of the network at high values of spread constant improves for a larger error goal. This may indicate the presence of noise in the data for ERE, which solverb is able to eliminate with an appropriate error goal. Fig. 17 shows variations in number of epochs and σ with the variation of error goal for a given spread constant when RBNN is designed with solverb. The number of neurons in the network is one more than the number of epochs. One expects that as the error goal increases the number of epochs becomes smaller and the network performs less accurately as in Figs. 17a and b.

When choosing an appropriate network the above-mentioned features must be considered. The performance of the constructed NN is best judged by comparing the prediction error as given in Eq. (8) for different networks. Using solverbe, networks are designed with varying spread constants and the one that yields the smallest error is selected. When solverb is used, networks are designed for different spread constants and error goals. The network that gives the smallest error for the test data is used. The details of the networks selected are discussed in later sections.

4.1.3. Evaluation of polynomial and NN for data processing

The polynomial- and NN-based RSM are constructed using the training data. The test data is then employed to select the best polynomial or NN. Specifically in polynomial-based RSM, the difference between the polynomial and the training data, as given by σ_a, is normally used to judge the performance of the fit. The additional use of the test data helps to evaluate the performance of different polynomials over design points not used during the training phase. This gives a complementary insight into the quality of the polynomial model over the design space. For example, Tables 4 and 5 compare the performance of different polynomials used to represent the two objective functions of the injector case, ERE and Q, for

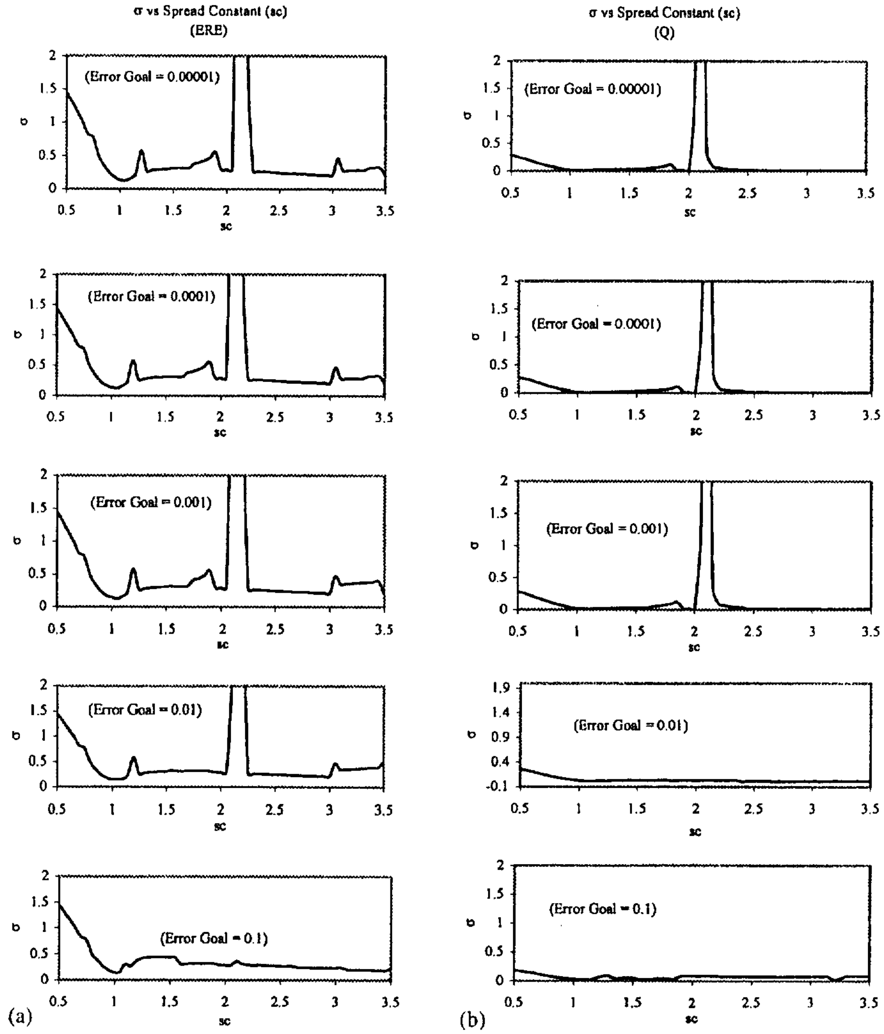


Fig. 16. Comparison of σ for different NN designed with solverb for shear co-axial gas–gas injector: (a) ERE (%) and (b) Q (Btu/in^2 s).

the shear co-axial injector. Starting with all the possible cubic terms in the model, revised models are generated by removing and adding terms. Similar kind of analysis is also done for the turbine case. The best polynomial is selected based on a combined evaluation between σ_a and σ.

For the NN, the test data helps evaluate the accuracy of networks with varying neurons in BPNN and varying spread constant in RBNN. Thus the test data are part of the evaluation process to help select the final NN. Based on the RSM or NN model, a search for optimum design is carried out using a standard, gradient-based optimization algorithm over the response surfaces represented by the polynomials and trained neural networks.

A reduced quadratic and an incomplete cubic response surfaces are used for the two objective functions. The first model in Table 4 and the sixth model in Table 5 are the selected cubic models for ERE and Q, respectively. There is no noticeable improvement amongst the remaining cubic models for ERE. For Q, the selected model is the best in terms of σ_a although there are other models with identical values of σ.

The radial basis networks designed with solverbe are the largest with 15 neurons in the hidden layer for ERE network and nine neurons for the Q network. Solverb designs a network for ERE with 14 neurons in the hidden layer and a network for Q with eight neurons. Compared to RBNN, BPNN has fewer neurons, the number of neurons in the hidden layer are eight and four for the ERE and Q networks, respectively. Details of the networks used are listed in Table 6. The spread constant used for RBNN and the error goal of the training data is also given in this table. The spread constant values are selected from the region where the performance of the

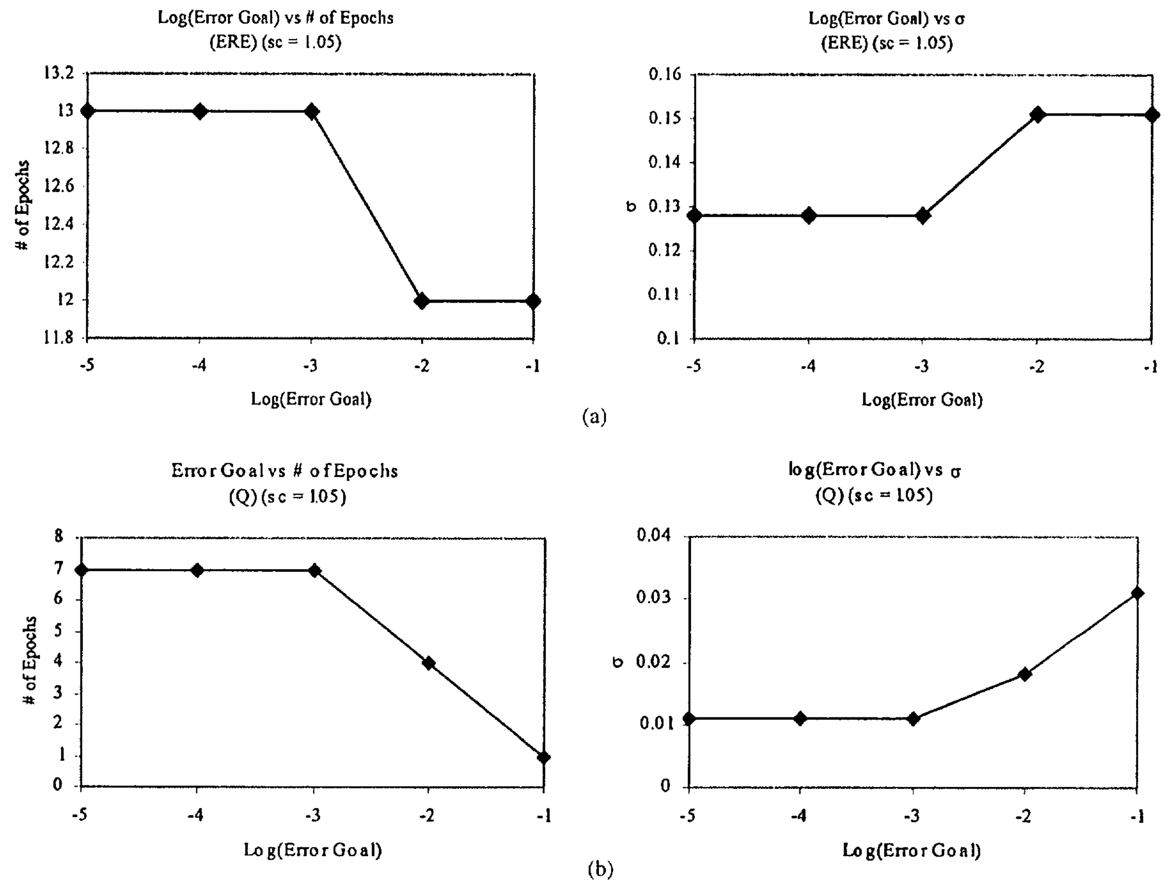


Fig. 17. Comparison of error goal versus number of epochs and σ for networks trained with solverb for shear co-axial gas–gas injector: (a) ERE (%), (b) Q (Btu/in^2 s).

Table 4
Different cubic polynomials for ERE (dependent variables: V_f/V_o and L_{comb}, 15 training points, 10 test points)

Model #	Coefficient = 0	Terms removed	Terms included	σ_a (%)	σ (%)
1	V_f/V_o^3			0.09	0.21
2	V_f/V_o^3	$V_f/V_o^2L_{comb}$		0.08	0.21
3		$V_f/V_o^2L_{comb}$, V_f/V_o^3		0.08	0.21
4		$V_f/V_o^2L_{comb}$, V_f/V_o^3	L_{comb}^4	0.09	0.21
5		$V_f/V_o^2L_{comb}$, V_f/V_o^3	L_{comb}^4, $V_f/V_o^2L_{comb}^2$	0.09	0.21
6		$V_f/V_o^2L_{comb}$, V_f/V_o^3	L_{comb}^4, $V_f/V_o^2L_{comb}^2$, V_f/V_oL_{comb}	0.10	0.21
T					

network is consistent with the variation of spread constant (Figs. 15 and 16). The error goal, in the case of solverb, is selected based on the network with the best performance for the ideal spread constant (Fig. 17).

The error in predicting the values of the objective function by different schemes is given in Table 7. Several observations can be readily made:

(1) Both NNs perform better than the RSM for this data set.
(2) Both solverbe and solverb are of comparable performance.
(3) The BPNN helps generate smaller networks and hence performs at par in comparison to RBNN.
(4) The cubic polynomial is more accurate than the quadratic one.

The various models generated are compared with test data shown in Figs. 18 and 19. The curves representing the NN predictions are closer to the data obtained

Table 5
Different cubic polynomials for Q (dependent variables: O/F and V_f/V_o, 9 training points, 4 test points)

Model #	Coefficient = 0	Terms removed	Terms included	σ_a (%)	σ (%)
1	$V_f/V_o^3, O/F^3$			5.58	2.23
2	O/F^3	V_f/V_o^2		5.58	2.09
3		$V_f/V_o^2, O/F^3$		5.58	2.09
4		$V_f/V_o^3, O/F^3$		5.58	2.23
5		$V_f/V_o^3, O/F^3, V_f/V_o^2$		3.96	2.09
6		$V_f/V_o^3, O/F^3, V_f/V_o^2$	$V_f/V_o^2, O/F^2$	5.58	2.09

Table 6
Neural network architectures used to design the model for shear co-axial injector element {sc = spread constant}

Scheme	# of layers	# of neurons in the hidden layer		# of neurons in the output layer		Error goal aimed for during training	
		ERE	Q	ERE	Q	ERE	Q
RBNN (Solverbe)	2	15	9	1	1	0.0 {sc = 3.25)	0.0 {sc = 1.20}
RBNN (Solverb)	2	14	8	1	1	0.001 {sc = 1.05}	0.001 {sc = 1.05}
BPNN	2	8	4	1	1	0.01	0.01

Table 7
rms-error in predicting the values of the objective function by various schemes for the shear co-axial injector element

Scheme	σ for ERE (%)	σ for Q (%)
RBNN (Solverbe)	0.20	1.40
RBNN (Solverb)	0.13	1.53
BPNN	0.18	0.83
Partial cubic RS	0.21	2.23
Quadratic RS	0.28	3.49

from the injector model than the RS thereby demonstrating that NN models are able to predict better than the RS. The BPNN performs as well as RBNN but tends to be flat. Due to its lower order, the quadratic polynomial is flat. The cubic polynomial is able to perform better than quadratic.

4.1.4. Polynomial-based RSM for design optimization

This case study is used to perform a complete comparative study between polynomial and NN-based RSM. The comparison is carried out in three ways. Firstly, the predictive capabilities of the different models are compared. Secondly, NN is used to increase the population of the design space, which is then used for mapping by polynomial-based RSM. Thirdly, polynomials and NN are used individually to represent the design space and help in the optimization of the design.

An optimization was done for three different ranges of the independent variables using the quadratic fit. The three cases analyzed differ only in the constraints implemented on the design parameters. The constraints are

Case 1: $4 \leqslant O/F \leqslant 6$, $4 \leqslant V_f/V_o \leqslant 6$, $L_{comb} \leqslant 7$.
Case 2: $4 \leqslant O/F \leqslant 6$, $5 \leqslant V_f/V_o \leqslant 7$, $L_{comb} \leqslant 7$.
Case 3: $4 \leqslant O/F \leqslant 6$, $6 \leqslant V_f/V_o \leqslant 8$, $L_{comb} \leqslant 7$.

The optimization is repeated using the cubic fits. The combinations of weighting factors for ERE, s, and for Q, t, are selected as (1, 10), (1, 1) and (10, 1) for these three cases. The optimum has been evaluated and tabulated for each case, as detailed in Tables 8–10. In this effort, injector element optimization means maximizing the performance, while minimizing heat flux and chamber length. The optimum value for V_f/V_o obtained on the cubic response surface is quite different than that found on the quadratic surface for some cases (these particular cases are noted in bold in Tables 8–10). Also, for selected cases where there are discrepancies between the quadratic and cubic results, the exact values from the injector model have been included in parentheses in the tables for comparison. In these cases, the cubic fit more closely matches the exact data than does the quadratic fit. Sample results for ERE plotted in Fig. 20a clearly show the data is better fit by the cubic surface for the case shown. Fig. 20b shows that the response surface predicted by cubic fit for Q has a noticeable dip that is completely missed by the quadratic fit. This discrepancy

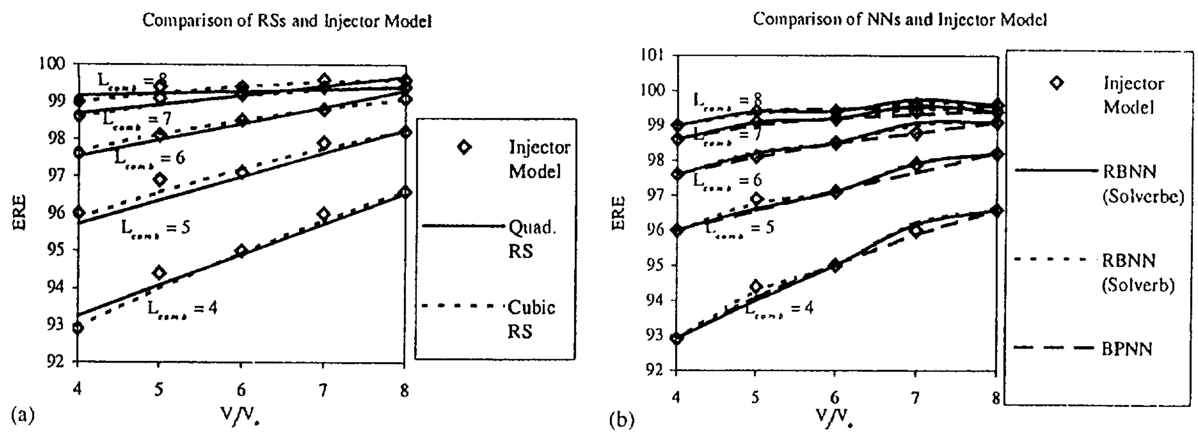


Fig. 18. Predictive capabilities of various models for ERE of the shear co-axial injector: (a) polynomial, (b) NN.

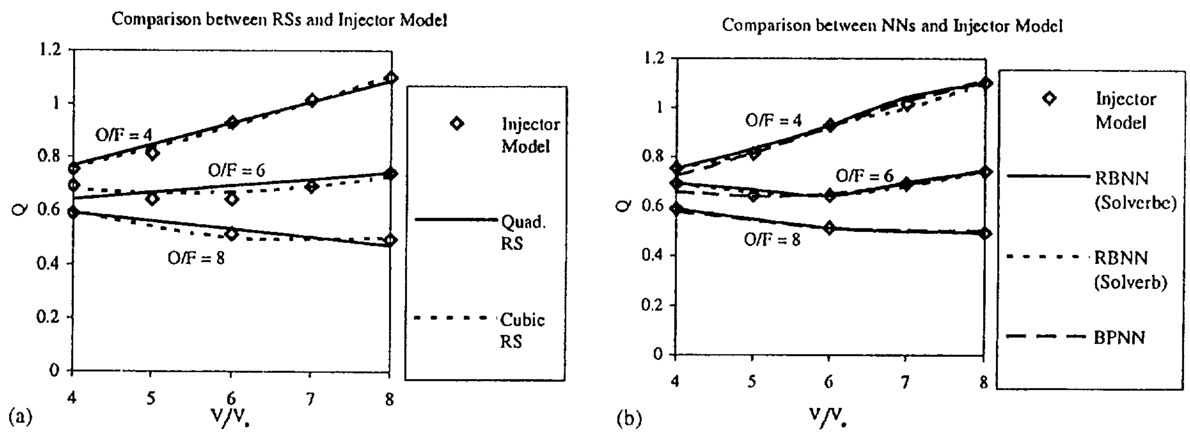


Fig. 19. Predictive capabilities of various models for Q of the shear co-axial injector: (a) polynomial, (b) NN.

Table 8

Optimum values obtained with cubic and quadratic for case 1 (constraints: $4 \leqslant O/F \leqslant 6$, $4 \leqslant V_f/V_o \leqslant 6$, and $L_{comb} \leqslant 7$) {values in the parenthesis are the exact response of the injector model}

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	**5.41**	7.0	99.02 (99.00)	0.664 (0.654)	6.0	**6.00**	7.0	99.17 (99.20)	0.669 (0.642)
1	1	6.0	6.00	7.0	99.15	0.669	6.0	6.00	7.0	99.17	0.669
10	1	6.0	6.00	7.0	99.15	0.669	6.0	6.00	7.0	99.17	0.669

results in the optimum for the cubic fit being considerably lower than that for the quadratic surface. The prediction from cubic fit agrees well with the exact data, which also has a dip for this specific case.

The injector model was also used to produce additional design points to assess the capability of the different response surfaces to match the exact data. In Figs. 21a and 22a, the actual data obtained from the injector model for all the design points has been shown. The cubic and quadratic response surfaces obtained based on the original data are also shown. The rms-error for predicting the new ERE data is 0.270 and 0.205 for the

Table 9
Optimum values obtained with cubic and quadratic for case 2 (constraints: $4 \leqslant O/F \leqslant 6$, $5 \leqslant V_f/V_o \leqslant 7$, and $L_{comb} \leqslant 7$) {values in the parenthesis are the exact response of the injector model}

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	**5.41**	7.0	99.02 (99.00)	0.664 (0.654)	6.0	**6.52**	7.0	99.31 (99.10)	0.684 (0.716)
1	1	6.0	**6.34**	7.0	99.21 (99.20)	0.674 (0.691)	6.0	**7.00**	7.0	99.42 (99.30)	0.702 (0.728)
10	1	6.0	7.00	7.0	99.32	0.690	6.0	7.00	7.0	99.42	0.702

Table 10
Optimum values obtained with cubic and quadratic for case 3 (constraints: $4 \leqslant O/F \leqslant 6$, $6 \leqslant V_f/V_o \leqslant 8$, and $L_{comb} \leqslant 7$)

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	**6.00**	7.0	99.15	0.669	6.0	**6.52**	7.0	99.31	0.684
1	1	6.0	**6.34**	7.0	99.21	0.674	6.0	**8.00**	7.0	99.67	0.753
10	1	6.0	8.00	7.0	99.42	0.728	6.0	8.00	7.0	99.67	0.753

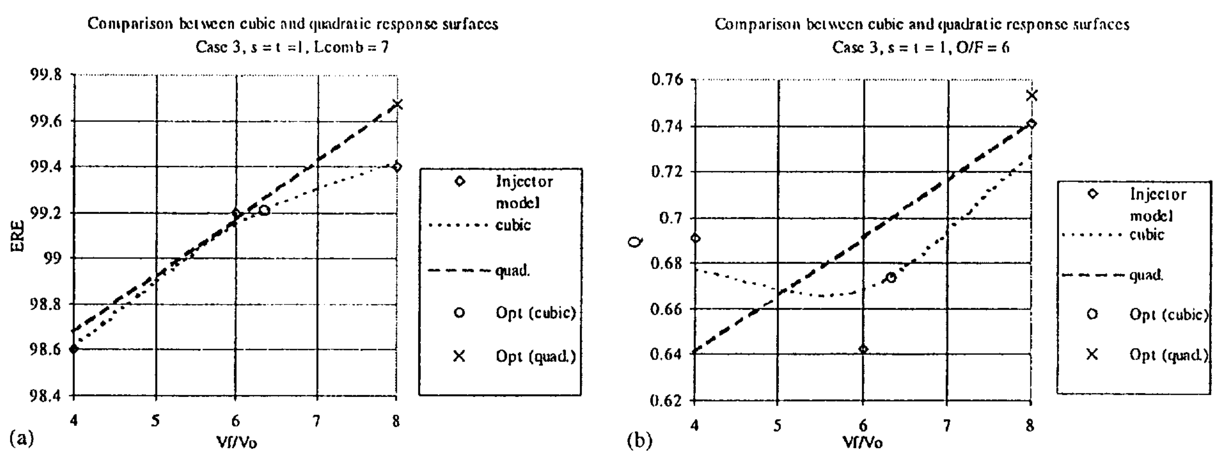


Fig. 20. Comparison between cubic and quadratic response surface for case 3 of: (a) ERE, (b) Q for the shear co-axial injector (NN and injector model data are the same points in the graph. Quadratic and cubic are predicted by polynomial-based RSM).

quadratic and cubic surfaces, respectively. For Q, it is 0.025 and 0.016 for the quadratic and cubic surfaces, respectively. Again, the performance of the cubic surface is superior to that of the quadratic surface.

4.1.5. *Radial basis neural networks (RBNN)*

Radial basis neural networks are trained by both solverbe and solverb for each injector design response, ERE and Q, using the original data set of 45 design points. Solverbe trained the network for ERE with an error to the order of 10^{-13}. The network trained by solverbe for Q has an error on the order of 10^{-16}. Both networks represent the respective design spaces essentially exactly. Solverb, with an error goal of 0.001, trained networks for both responses to represent the original data set adequately. Since the size of the data set considered for training the network is fairly modest, the number of neurons generated by solverbe is also small. Solverb would have been suited better for a larger data set where a reduction in the number of neurons might

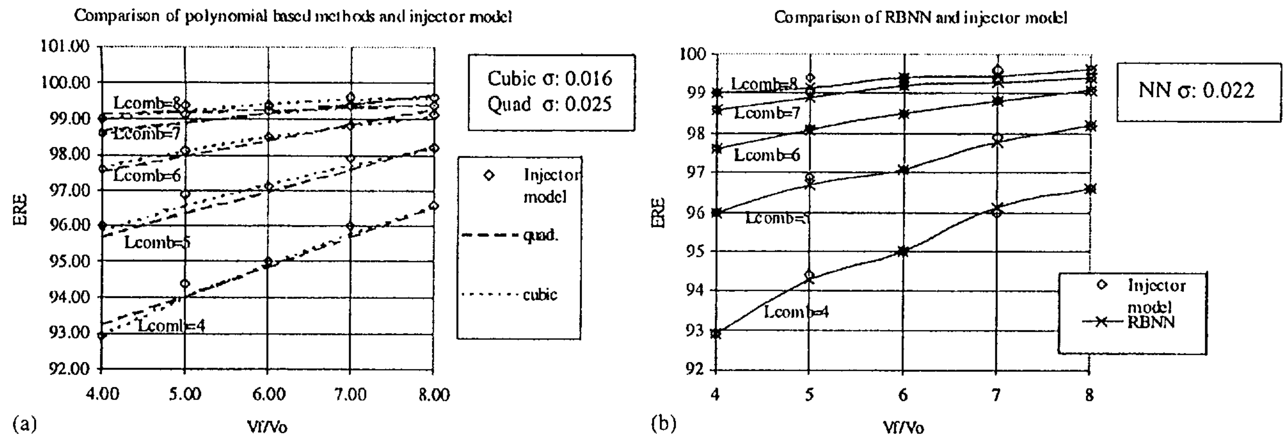


Fig. 21. Assessment of predictive capabilities of: (a) polynomial based method, (b) RBNN for ERE of the shear co-axial injector (15 training/mapping points 10 test points).

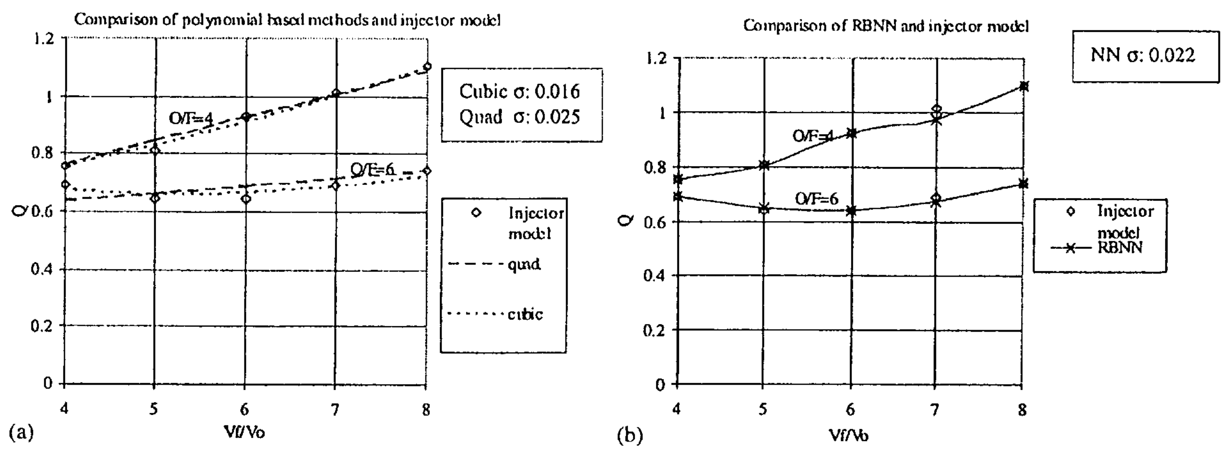


Fig. 22. Assessment of predictive capabilities of: (a) polynomial based methods and (b) RBNN for Q (9 training/mapping points, 4 test points) for shear co-axial injector design.

have appreciably reduced the computation time. The networks trained using solverbe have been used for this study. The ability of the RBNN to fit the design data and to generate additional data for constructing a more accurate response surface is discussed in the following sections.

(i) *Comparison between Solverbe and Solverb*: Since Solverbe trains with the same number of neurons (45 in this case) as data points, as noted above, it fits the training data set with negligible error. However, it can also create erratic behavior since it makes no attempt to filter noise generated by excess neurons in the network. Solverb, on the other hand, tends to reduce the potential for noise by controlling the number of neurons in the network. Table 11 shows that in the present effort, for the spread constant value of 1.00, Solverb performs slightly better than Solverbe based on the nominal error measure. However, when judged by the level of errors associated, both RBNNs are satisfactory from a practical standpoint. As expected, Solverb uses fewer neurons than Solverbe; in this case three less. It should be noted that, as investigated in detail by Papila et al. [37], indicates the relative performance between Solverb and Solverbe is case dependent.

(ii) *Comparison of RBNN predictions with polynomial-based RSM*: Figs. 21b and 22b show that the RBNN trained by Solverbe is able to more accurately generate additional design data than either quadratic or cubic polynomial (shown for comparison in Figs. 21a and 22a). In Fig. 21a, the ERE surface trained with the original data set is shown. The 10 extra design points calculated with the injector model for V_f/V_o of 5.00 and 7.00 are shown. The ability of the RBNN to accurately generate new design data can be seen by comparing the fit for ERE in Fig. 21b to that for the polynomials in Fig. 21a. The RBNN trains the network with more flexibility and

Table 11
The rms-error in the prediction of ERE and Q for different values of spread constant. The error goal used for Solverb is 0.001

Sc	Solverbe rms-error (ERE)	Solverbe rms-error (Q)	Solverb rms-error (ERE)	Solverb rms-error (Q)	Solverb No. of neurons
0.50	1.493	0.179	1.733	0.287	44
0.75	0.745	0.135	0.675	0.135	44
1.00	**0.152**	**0.022**	**0.153**	**0.017**	**42**
1.05	**0.190**	**0.011**	**0.128**	**0.012**	**44**
1.25	0.316	0.010	0.267	0.022	44
1.50	0.336	0.022	0.309	0.030	44
1.75	0.369	0.022	0.310	0.021	44
2.00	0.308	0.016	0.296	0.019	41
2.25	0.279	0.020	1.846	0.045	43
2.50	0.325	0.017	0.744	0.025	43

learns the data trend, whereas polynomials provide only an approximate fit on the given data. Regarding the rms-error, σ, for ERE, it is 0.152 for RBNN predictions as compared to the values of 0.270 and 0.205 for quadratic and cubic surfaces, respectively. The four extra design points generated for Q, also at V_f/V_o of 5.00 and 7.00 are shown compared to the polynomial in Fig. 22a and RBNN in Fig. 22b. The rms-error in the case of Q is 0.022 for RBNN as compared to 0.025 and 0.016 for quadratic and cubic surfaces, respectively. Here the performance of the RBNN is better than the quadratic but slightly poorer than the cubic fit. Examination of Table 11 indicates it may be possible that using Solverb with a spread constant of 1.05 could further reduce the rms-error for Q. However, the errors for Q are low enough that further reduction may not be practical.

4.1.6. RBNN-enhanced polynomial-based response surface

Additional design points generated by the RBNN are added to the original data set to form the enhanced data set. This enhanced data set is used for further analysis to evaluate the performance of the RSM with the larger number of design points. The enhanced data set for ERE has 15 points from the injector model and 10 from the RBNN, for a total of 25 points. The enhanced data set for Q has 9 points from the injector model and 4 from the RBNN, for a total of 13 points. The entire optimization analysis was redone with the enhanced data set. On this enhanced data set, the full quadratic response surface seems already appropriately constructed and invoking the statistical analysis generates no reduced model. With the added data in the enhanced data set, it is now possible to obtain a fit for ERE that is fourth order in V_f/V_o and fourth order in L_{comb}. Q can now be fit with a cubic in V_f/V_o and a quadratic in O/F. This is now possible since a combination of 3 different values of O/F, 5 different values of V_f/V_o and 5 different values of L_{comb} are available.

(i) *Comparison of fits with the original response surfaces*: Comparison of the enhanced response surfaces with the original response surfaces indicates that the extra data produced with the RBNN generally improves the quality of the curve fit. The adjusted rms-error for ERE on the original set is 0.211 and 0.083 for quadratic and cubic fits, respectively. On the enhanced data set, it is 0.179 and 0.100 for the quadratic and cubic fits, respectively. The slight increase in the error in the case of the cubic fit may be due to noise related to the over-sensitivity of the polynomial. However, this phenomenon may reflect the fact that the level of the rms-error is low enough in either case so that no further improvement is accomplished. The adjusted rms-error for Q with the original set is 0.039 and 0.040 for the quadratic and cubic fits, respectively. On the enhanced set it was 0.027 and 0.026 for the quadratic and cubic, respectively. With the exception of the cubic fit for ERE, the fits from the enhanced surface are improved over those from the original surface. Also, when optimum design points are examined, there is less difference between the quadratic and cubic fits on the enhanced surfaces than there is on the original surfaces.

(ii) *Comparison of optimal design points*: The analysis for the three cases of optimization over the same three ranges of independent variables has been re-done. The results of the optimization on surfaces generated from the enhanced data set are tabulated in Tables 12–14. The predicted optimal design points using cubic and quadratic fits are generally close to each other. They are closer to each other on the enhanced data set than on the surfaces generated using the original data set. One case where the cubic and quadratic optimum points are somewhat different is analyzed further. The results shown in Fig. 23 confirm the optimum value of velocity ratio on the quadratic fit to be lower than the cubic fit in this case. Given the weightings of 1.0 for ERE and 10.0 for Q, the optimizer has selected the minimum of Q for both fits. Since the curves exhibit different minimum points, the weightings force the selection of different optimum

Table 12
Optimum values obtained with cubic and quadratic for case 1 (enhanced data set) (constraints: $4 \leqslant O/F \leqslant 6$, $4 \leqslant V_f/V_o \leqslant 6$, and $L_{comb} \leqslant 7$) {cf. with Table 8}

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	**5.54**	7.0	99.02 (98.90)	0.654 (0.658)	6.0	**5.01**	7.0	98.96 (98.70)	0.644 (0.664)
1	1	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658
10	1	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658

Table 13
Optimum values obtained with cubic and quadratic for case 2 (enhanced data set) (constraints: $4 \leqslant O/F \leqslant 6$, $5 \leqslant V_f/V_o \leqslant 7$, and $L_{comb} \leqslant 7$) {cf. with Table 9}

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	**5.54**	7.0	99.02 (98.90)	0.654 (0.658)	6.0	**5.01**	7.0	98.96 (98.70)	0.644 (0.664)
1	1	6.0	**6.33**	7.0	99.18 (99.10)	0.663 (0.666)	6.0	**6.04**	7.0	99.26 (99.20)	0.659 (0.642)
10	1	6.0	7.00	7.0	99.30	0.681	6.0	7.00	7.0	99.46	0.693

Table 14
Optimum values obtained with cubic and quadratic for case 3 (enhanced data set). (constraints: $4 \leqslant O/F \leqslant 6$, $6 \leqslant V_f/V_o \leqslant 8$, and $L_{comb} \leqslant 7$) {cf. with Table 10}

		Cubic					Quadratic				
W_{ERE} (s)	W_Q (t)	O/F	V_f/V_o	L_{comb}	ERE	Q	O/F	V_f/V_o	L_{comb}	ERE	Q
1	10	6.0	6.00	7.0	99.12	0.658	6.0	6.00	7.0	99.25	0.658
1	1	6.0	**6.33**	7.0	99.19	0.663	6.0	**6.04**	7.0	99.26	0.659
10	1	6.0	8.00	7.0	99.42	0.725	6.0	7.95	7.0	99.57	0.746

points. As already discussed, for the polynomial fits on the RBNN-enhanced data sets, the errors of both quadratic and cubic polynomials are more comparable than in the original analysis. At the upper limit of the design space for combustor length, the ERE curves tend to flatten out. This causes some difficulty in locating the optimum and may cause more noticeable differences between the different polynomials. However, different optimal designs selected by different polynomials under such a circumstance are not significant since these yield very similar injector performance.

The optimum solution obtained from various schemes is shown in Table 15 and Figs. 24 and 25. The aim is to maximize ERE and minimize Q. The trend of the objective functions in the design space is monotonic, hence every model is able to select identical optimum design for the given constraints. The flatness of the quadratic polynomial results in less accurate values of the objective function for the optimum design. However, the cubic polynomial, while more flexible than quadratic, is not consistently better in predicting the optimal design point. For example, a V_f/V_o constraint of 4, the quadratic polynomial is more accurate but for higher values of V_f/V_o the cubic polynomial is more accurate. In contrast, the NN models are able to perform consistently well. Since the optimum design happens to be the same as one of the training points, solverbe is able to predict the values of the objective function accurately. Solverb performs equally well, illustrating the capability of performance with fewer neurons. Performance of BPNN is not

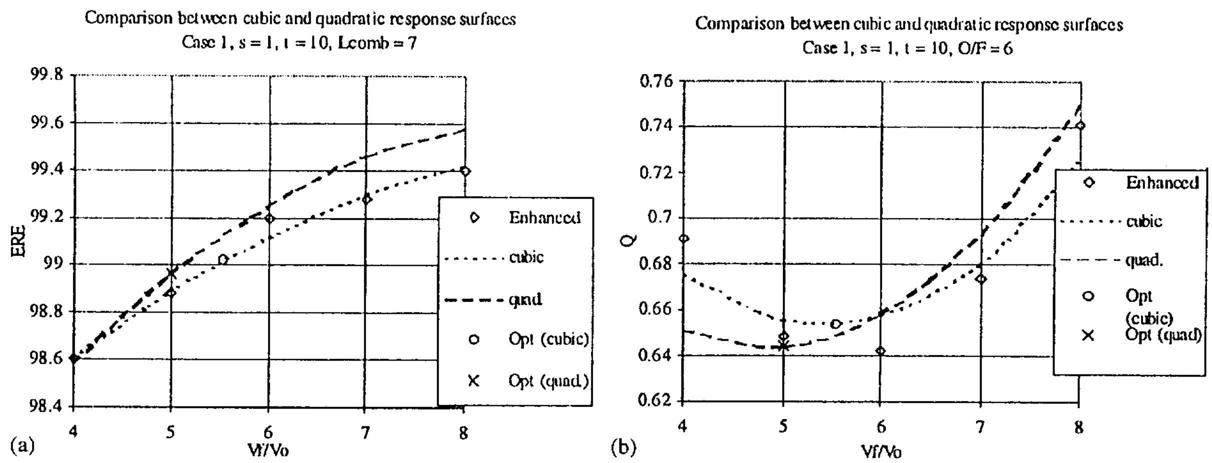


Fig. 23. Assessment of performance of cubic and quadratic response surfaces for case 1 of: (a) ERE for the shear co-axial injector (25 training/mapping points), (b) Q (13 training/mapping points) {Enhanced set includes Injector model data and RBNN predicted data. Quadratic and cubic are predicted by polynomial-based RSM}.

Table 15
Optimal solutions for fixed values of V_f/V_o and given range of O/F and L_{comb} obtained with NN and polynomial-based RSM schemes for the shear co-axial injector element (constraints: $4 \leqslant O/F \leqslant 8$, $4 \leqslant L_{comb} \leqslant 7$) (error given in parenthesis for each prediction is in %)

V_f/V_o	Scheme	O/F	L_{comb} (in)	ERE (%)	Q (Btu/in^2-s)
4	RBNN (Solverbe)	8.0	7.0	98.60 (0.00)	0.588 (0.00)
	RBNN (Solverb)	8.0	7.0	98.60 (0.00)	0.588 (0.00)
	BPNN	8.0	6.9	98.64 (0.14)	0.578 (1.70)
	Partial cubic RS	8.0	7.0	98.61 (0.01)	0.595 (1.19)
	Quadratic RS	8.0	7.0	98.67 (0.07)	0.591 (0.51)
	Model	8.0	7.0	**98.60**	**0.588**
	Model	8.0	6.9	**98.50**	**0.588**
6	RBNN (Solverbe)	8.0	7.0	99.20 (0.00)	0.512 (0.00)
	RBNN (Solverb)	8.0	7.0	99.20 (0.00)	0.512 (0.00)
	BPNN	8.0	7.0	99.18 (0.02)	0.513 (0.20)
	Partial cubic RS	8.0	7.0	99.15 (0.05)	0.499 (2.54)
	Quadratic RS	8.0	7.0	99.17 (0.03)	0.531 (3.71)
	Model	8.0	7.0	**99.20**	**0.512**
8	RBNN (Solverbe)	8.0	7.0	99.40 (0.00)	0.493 (0.00)
	RBNN (Solverb)	8.0	7.0	99.40 (0.00)	0.493 (0.00)
	BPNN	8.0	7.0	99.41 (0.01)	0.500 (1.42)
	Partial cubic RS	8.0	7.0	99.42 (0.02)	0.500 (1.42)
	Quadratic RS	8.0	7.0	99.67 (0.27)	0.471 (4.46)
	Model	8.0	7.0	**99.40**	**0.493**

as satisfactory as suggested in Table 7. For lower constraints of V_f/V_o, it performs poorly but for higher values of V_f/V_o it is good. This may be due to the selection of fewer neurons in the hidden layers of the networks. Overall, it is still better than the polynomial-based RSM and demonstrates the flexibility of NN over polynomials.

As stated by Papila et al. [37], when it comes to choosing between NN and polynomials, polynomials are easy to compute. The number of coefficients might be numerous but the linearity of the system expedites the process of coefficient evaluations. This is also the reason RBNN train fast. On the other hand, the weights of BPNN are evaluated through a nonlinear optimization, which slows the training process. Of all the NN presented here, the one designed with the help of solverbe is the fastest to train since the values of the weights are set to

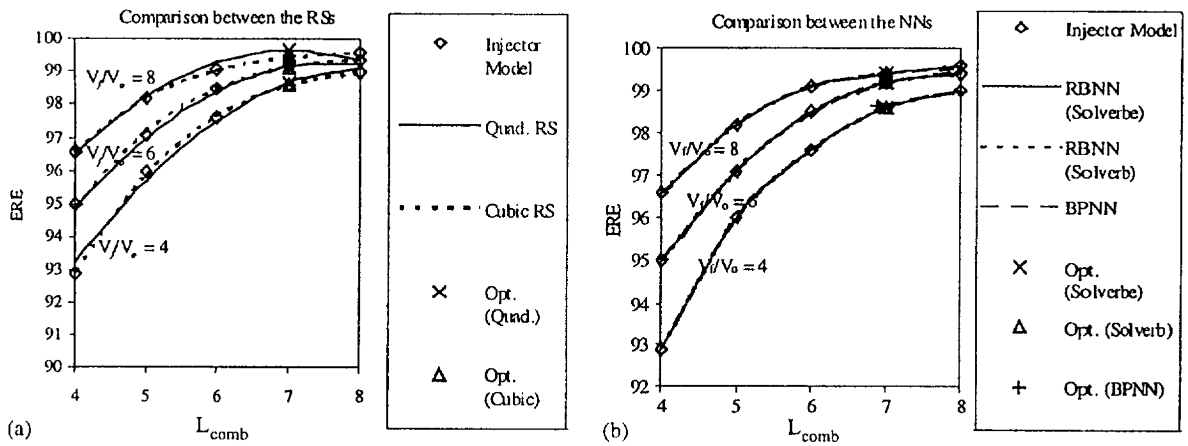


Fig. 24. Performance of various models for ERE of the shear co-axial injector: (a) RSs, (b) NNs.

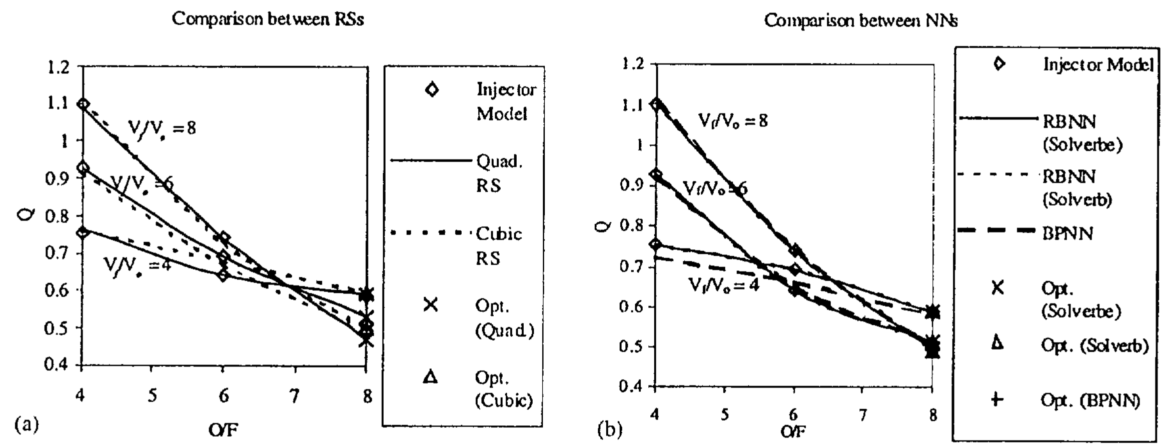


Fig. 25. Performance of various models for Q of the shear co-axial injector: (a) RSs, (b) NNs.

values of the input dependent variables. Solverb trains with the addition of one neuron at a time with weights similar to the input and hence is slower.

4.2. Impinging injector element

4.2.1. Polynomial fits

In [55], method *i* uses the polynomial based RSM to find optimal values of ERE, Q_w, Q_{inj}, W_{rel} and C_{rel} for acceptable values of ΔP_o, ΔP_f, L_{comb} and α. The approach of RSM is to perform a series of experiments, or numerical analyses, for a prescribed set of design points, and to construct a response surface of the measured quantity over the design space. In the present context, the five responses of interest are ERE, Q_w, Q_{inj}, W_{rel} and C_{rel}. The design space consists of the set of relevant design variables ΔP_o, ΔP_f, L_{comb} and α.

(i) *Individual polynomial models*: When JMP is used to analyze the 163 design points, five individual full response surfaces for the variables in the design space are approximated by quadratic polynomials that contain 15 terms each. Using the *t*-statistics approach noted above and detailed in [55], unnecessary terms in each equation can be eliminated to give the reduced quadratic surfaces. The reduced response surfaces indicates that the equations reflect the functionality used to construct the models for the dependent variables.

(ii) *Joint response surfaces*: In the current article, it is desirable to maximize ERE and while simultaneously minimizing Q_w, Q_{inj}, W_{rel} and C_{rel}. Therefore, composite response surface for the present case is given by

$$D = (d_{ERE} d_{Q_w} d_{Q_{inj}} d_{W_{rel}} d_{C_{rel}})^{1/5}. \tag{30}$$

4.2.2. Optimization results and discussion

Three sets of results are presented below to demonstrate the capability of method *i* for the current injector design. These three examples illustrate the effect of each

Table 16
Effect of each variable on the optimization of impingement co-axial element—optimal designs for original constraints and equal weights

Independent variable	Constraints	Results Case 1	Results Case 2	Results Case 3	Results Case 4
ΔP_o	100–200	183	183	179	100
ΔP_f	100–200	100	132	149	100
L_{comb}	2–8	8.0	8.0	6.6	6.5
α	15–50	33.1	18.9	22.3	24.0
Dependent variable	Desirability limits	ERE & Q_w	ERE, Q_w, $H_{impinge}$	ERE, Q_w, $H_{impinge}$, W_{rel}	ERE, Q_w, $H_{impinge}$, W_{rel}, C_{rel}
ERE	95.0–99.9	99.9	98.3	98.0	98.0
Q_w	0.7–1.3	0.74	0.76	0.79	0.86
$H_{impinge}$	0.2–1.0	—	0.75	0.61	0.63
W_{rel}	0.9–1.2	—	—	1.1	1.1
C_{rel}	0.7–1.1	—	—	—	0.93

variable on the optimum design, the trade-offs between life and performance issues, and the effect on the design of extracting the last increment of performance.

(i) *Effect of each variable on the design using original constraints and equal weights*: The results in this section were obtained by building the joint response surface with the addition of one dependent variable at a time. The results are shown in Table 16. Since current non-optimizer-based design methods yield high-performing injector elements, simply maximizing the ERE is not a challenge. Accordingly, the initial results (Case 1) are obtained with a joint ERE and Q_w response surface. The results in Case 2 have the impingement height added, Case 3 adds the relative chamber weight and the relative cost is added in Case 4. All results are obtained using the original independent variable constraints and all dependent variables have equal weights of one. The results for Case 1 show that ERE is at its maximum and Q_w is very near its minimum desirability limit. Minimizing Q_w requires a small ΔP_f relative to ΔP_o as evidenced by the values of 100 and 183 psi, respectively. Maximum ERE values are found at the longest chamber length, $L_{comb} = 8$ inc. Even with the relatively high value of 183 psi for ΔP_o and low value of ΔP_f of 100 psi, ERE is maximized to 99.9% with an impingement half-angle of 33.1°.

Addition of the impingement height to Case 2 to model the injector face heat flux, Q_{inj}, forces α lower to increase $H_{impinge}$ and decrease Q_{inj}. This decrease in the radial component of the fuel momentum has an adverse affect on ERE. This effect is mitigated to a degree by increasing the ΔP_f from 32 to 132 psi. ERE is still reduced by 1.6%. Also, the increase in ΔP_f causes increased penetration of the fuel jet, which results in a slightly higher Q_w.

Case 3 adds the relative combustor weight to the list of dependent variables modeled. Since W_{rel} is only a function of L_{comb}, minimizing W_{rel} shortens the combustor length from 8 to 6.6 in. The shorter L_{comb} tends to lower ERE. This effect is offset to a large degree by increases in ΔP_f and α, both of which increase the radial component of the fuel momentum. The increase in ΔP_f also causes a slight increase in Q_w. The increase in α causes a significant decrease in $H_{impinge}$, which increases the injector face heat flux. Finally, the relative cost of the injector is added in Case 5. Since C_{rel} is only a function of propellant pressure drops, both ΔP_o and ΔP_f are driven to their respective minimum values. This and a slight increase in α allow ERE to be maintained at 98%, even with a slight decrease in L_{comb}. The largest effect of this fairly dramatic decrease in propellant pressure drops is on Q_w. Even though the values for ΔP_o and ΔP_f fell, ΔP_f increased relative to ΔP_o causing Q_w to increase by almost 9%. Impingement height and relative combustor weight are essentially unchanged.

Although several of the variables included in this exercise are qualitative, an important conclusion can still be drawn. The sequential addition of dependent variables to an existing design results in changes to both the independent and dependent variables in the existing design. The direction and magnitude of these changes depends on the sensitivity of the variables, but the changes may well be significant. The design in Case 4 is quite different that the one in Case 1. Consideration of a larger design space results in a different design — the sooner the additional variables are considered, the more robust the final design will be.

(ii) *Emphasis on life and performance issues using original constraints and unequal weights*: The purpose of this section is to illustrate the effect of emphasizing certain design criterion on the optimization process. Method *i* allows this emphasis via the weights applied to the desirability functions in the joint response surface. The results shown in Table 17 facilitate the illustration. Case 1 (baseline) results are repeated from Case 4 in this

Table 17
Effect of emphasizing & life & performance issues on the optimization of impingement co-axial injector element—optimal designs for original constraints and modified weights

Independent variable	Constraints	Results Case 1	Constraints	Results Case 2	Constraints	Results Case 3
ΔP_o	100–200	100	100–200	158	100–200	100
ΔP_f	100–200	100	100–200	100	100–200	137
L_{comb}	2–8	6.5	2–8	7.7	2–8	5.2
α	15–50	24.0	15–50	15.0	15–50	36.0
Dependent variable	Baseline variable weight		Life variable weight		Thrust/weight variable weight	
ERE	1	98.0	1	96.7	5	99.1
Q_w	1	0.86	5	0.75	1	0.95
$H_{impinge}$	1	0.63	5	0.94	1	0.32
W_{rel}	1	1.10	1	1.14	5	1.05
C_{rel}	1	0.93	1	0.97	1	0.95

table where the entire design space is considered with the original constraints and equal weights for the dependent variables. The results in Case 2 column are obtained by emphasizing the minimization of the wall and injector face heat fluxes. Desirability functions for both of these variables are given a weight of five. Since lower heat fluxes tend to increase component life, weighting these two variables is equivalent to emphasizing a life-type issue in the design. As expected, α is decreased to increase $H_{impinge}$, thus decreasing Q_{inj}. Since the fuel pressure drop is already at the minimum, the oxidizer pressure drop is increased by 58% to decrease Q_w. Both of these changes tend to decrease ERE. While ERE does decrease, the effect is somewhat mitigated by an increase in L_{comb}. The increases in L_{comb} and ΔP_o cause increases in W_{rel} and C_{rel}, respectively. The emphasis on life extracts the expected penalty on performance. Additionally, for the current model, there are also weight and cost penalties.

The results for Case 3 are obtained by emphasizing maximization of ERE and minimization of W_{rel} with desirability weightings of five. Increased weighting for these two variables is equivalent to emphasizing a thrust to weight goal for the injector/chamber. The relative chamber length is shortened to lower W_{rel}. ERE is maximized by increasing the radial momentum of the fuel jet. Both ΔP_f and α are increased to accomplish ERE maximization. As noted earlier, increasing ΔP_f and α lead to increased wall and injector heat fluxes, respectively. Table 17 indicates that to be the case here. For this case, emphasis on thrust and weight tend to have an adverse affect on both Q_w and Q_{inj}. Relative cost, for the current model, is not significantly affected.

(iii) *Extraction of last performance and weight increments (modified constraints and unequal weights)*: Here, the high marginal cost of realizing the last increment of thrust to weight is shown. This section illustrates the capability to modify the constraints on the independent variables and use unequal weights on the dependent variables at the same time. The results for Case 3 in Table 17 are carried over to Case 1 in Table 18 as the baseline for this example. Here the original constraints are used but increased weights have been applied to emphasize ERE and W_{rel}. Cases 2 and 3 modify the constraints on the propellant pressure drops, raising the minimum pressure drop from 100 to 150 psi. For Case 2, both ΔP_o and ΔP_f are now at the minimum level for the modified constraints. L_{comb} is increased slightly to maintain ERE. The decrease of ΔP_f relative to ΔP_o causes a decrease in Q_w. The slightly higher-pressure drops also cause C_{rel} to increase somewhat. Other variables are not changed appreciably.

For Case 3, ERE and W_{rel} are further emphasized by increasing their desirability weights to 10 while decreasing the other weights to 0.1. L_{comb} is shortened to respond to the increased emphasis on weight minimization. Maintaining the high level of ERE requires large increases in ΔP_f and α to increase the radial component of the fuel jet momentum. The increase in ΔP_f causes over-penetration of the fuel jet, which results in an increase in wall heat flux. The large increase in α yields the expected decrease in $H_{impinge}$, which increases the injector face heat flux. The additional emphasis on ERE and C_{rel} yields essentially no increase in ERE in this range of ΔP, although a small weight savings is seen. These marginal improvements are offset by fairly large increases in C_{rel} and Q_{inj}.

4.3. Swirl co-axial injector element

Two sets of results are presented below to demonstrate the capability and flexibility of method *i* for the current

Table 18
Effects of realizing the last increments of performance & weight on the optimization of impingement co-axial injector element —optimum designs for modified constraints and unequal weights

Independent variable	Original constraints	Results Case 1	Modified ΔP constraints	Results Case 2	Modified ΔP constraints	Results Case 3
ΔP_o	100–200	100	150–200	150	150–200	150
ΔP_f	100–200	137	150–200	150	150–200	200
L_{comb}	2–8	5.2	2–8	5.4	2–8	4.4
α	15–50	36.0	15–50	35.6	15–50	44.8
Dependent variable	Variable weight (5:1)		Variable weight (5:1)		Variable weight (100:1)	
ERE	5	99.1	5	99.0	10	99.1
Q_w	1	0.95	1	0.84	0.1	0.95
$H_{impinge}$	1	0.32	1	0.31	0.1	0.21
W_{rel}	5	1.05	5	1.05	10	1.01
C_{rel}	1	0.95	1	1.00	0.1	1.07

Table 19
Effect of each variable on the optimization of swirl co-axial injector element—optimal designs for original constraints and equal weights

Independent variable	Constraints	Results Case 1	Results Case 2	Results Case 3	Results Case 4	Results Case 5
ΔP_o	100–200	200	200	200	200	104
ΔP_f	20–200	41	41	42	47	20
L_{comb}	2–8	7.2	7.2	7.6	3.2	3.4
Θ	30–90	81	81	37	47	44
Dependent variable	Desirability limits	ERE	ERE & Q_w	ERE, Q_w, Q_{inj}	ERE, Q_w, Q_{inj}, W_{rel}	ERE, Q_w, Q_{inj}, W_{rel}, C_{rel}
ERE	92.3–99.0	98.5	98.5	97.2	96.0	95.7
Q_w	0.596–0.647	0.596	0.596	0.596	0.596	0.596
Q_{inj}	6.95–36.59	26.8	26.8	9.1	12.0	10.5
W_{rel}	0.900–1.154	1.13	1.13	1.14	0.97	0.98
C_{rel}	0.73–1.42	0.98	0.98	0.81	0.84	0.76

injector design. These examples illustrate the effect of each variable on the optimum design and the trade-offs between life and performance issues.

4.3.1. Effect of each variable on element design

The results in this section were obtained by building the joint response surface with the addition of one dependent variable at a time. The results are shown in Table 19. Case 1 seeks the maximum performance without regard to the effect on the other dependent variables. ERE is a fairly strong function of L_{comb}—longer chamber lengths allow more residence time for the propellant to mix and burn. The effect of Θ on ERE is strongest at low values of Θ. ERE increases with increasing Θ until about $\Theta = 80°$ and then fall off slightly due to the competing influences noted earlier. These competing influences also cause the effect of both pressure drops on ERE to be somewhat flat, although since ΔP_o affects more variables, its influence is slightly stronger. Maximum performance is found at high values of ΔP_o, Θ, and L_{comb} and at low values of ΔP_f. This trend is consistent with other works for similar injector elements. The predicted optimal value of 98.5 is indeed the highest predicted by this model.

The objective of Case 2 is to simultaneously maximize ERE and minimize Q_w. Table 19 shows that the exact same design point was chosen as for Case 1. Usually, the design, which yields the maximum ERE, also produces a high wall heat flux. That is not the case here; this issue has already been noted. The minimum Q_w is found in the region of high ΔP_o and low ΔP_f. In this area, Q_w is almost independent of Θ. Hence, the minimum Q_w can still be found for a high value of Θ required to maximize

Table 20
Effect of emphasizing life and performance issues on the optimization of swirl co-axial injector element

Independent variable	Constraints	Results baseline	Constraints	Results Case 1	Constraints	Results Case 2
ΔP_o	100–200	104	100–200	200	100–200	200
ΔP_f	20–200	20	20–200	32	20–200	44
L_{comb}	2–8	3.4	2–8	3.6	2–8	2.9
Θ	30–90	44.0	30–90	30.0	30–90	72.0
Dependent variable	Baseline variable weight		Life variable weight		Thrust/weight variable weight	
ERE	1	95.7	1	95.3	10	96.7
Q_w	1	0.596	5	0.596	1	0.596
Q_{inj}	1	10.5	10	6.9	1	22.6
W_{rel}	1	0.98	1	0.99	2	0.96
C_{rel}	1	0.76	1	0.79	1	0.94

ERE. It should be noted that in the low ΔP_o, high ΔP_f region, Q_w is a function of Θ. Here, as Θ is increased, Q_w increases since the larger swirl angle forces d_o to increase and thus decrease V_o. In the Calhoon et al. [57] model, this reduction in GO_2 momentum causes an increase in Q_w.

The requirement to minimize Q_{inj} is added in Case 3. In order to minimize Q_{inj}, the swirl angle is decreased from 81 to 37°, thus reducing the injector face heat flux by approximately a factor of 3. This decrease in Θ also lowers ERE which forces use of a longer chamber to offset some of the loss. Still, ERE is reduced by over one percent.

Case 4 considers the desire to minimize the chamber weight, W_{rel}, in addition to maximizing ERE and minimizing Q_w and Q_{inj}. Since W_{rel} depends only on L_{comb}, the chamber length is shortened by over half. The weight is reduced, but so is ERE. To mitigate the adverse effect on ERE, Θ is increased by almost 10°, simultaneously increasing Q_{inj}. ERE drops again by over a percent, while Q_w remains constant.

Finally, minimizing the injector cost, C_{rel}, is added in Case 5. Decreasing each pressure drop approximately a factor of 2 lowers the relative injector cost. Decreasing ΔP_f results in a larger fuel gap and decreasing ΔP_o allows for a larger swirl slot. These factors combine to lower the cost by almost 10%.

Although several of the variables included in this exercise are qualitative, an important conclusion can still be drawn. The sequential addition of dependent variables to an existing design results in changes to independent and dependent variables in the existing design. The direction and magnitude of these changes depends on the sensitivity of the variables, but the changes may well be significant. The design in Case 5 is quite different than the one in Case 1. Consideration of a larger design space may result in a significantly different design—the sooner the additional variables are considered, the more robust the final design.

4.3.2. Emphasis on life and performance issues

Method i allows this emphasis via the weights applied to the desirability functions in the joint response surface. The set of results shown in Table 20 facilitate the illustration. The baseline results Table 20 (repeated from Case 5 in Table 19) consider the entire design space using the original constraints and equal weights for the dependent variables. The results are obtained by emphasizing the minimization of the wall and injector face heat fluxes for Case 1. Desirability functions for both of these variables are given increased weights (5 and 10, respectively). Since lower heat fluxes tend to increase component life, weighting these two variables is equivalent to emphasizing a life-type issue in the design. Since Q_w is already at its minimum value, it remains fixed. As expected, Θ is decreased, which decreases the value of Q_{inj} by almost 35%. The lower value of Θ also produces a lower ERE. Both propellant pressure drops and the combustor length are increased to mitigate the drop in ERE. The increases in L_{comb} and ΔP_f cause increases in W_{rel} and C_{rel}, respectively. The emphasis on life extracts the expected penalty on performance. Additionally, for the current model, there are also slight weight and cost penalties.

The results for Case 2 are obtained by emphasizing maximization of ERE and minimization of W_{rel} with desirability weightings of 10 and 5, respectively. Increased weighting for these two variables is equivalent to emphasizing a thrust to weight goal for the injector/chamber. The relative chamber length is shortened to slightly lower W_{rel}. ERE is maximized by increasing the GO_2 swirl angle by a factor of almost 2.5 and also increasing ΔP_f by over 35%. The value of ERE increases

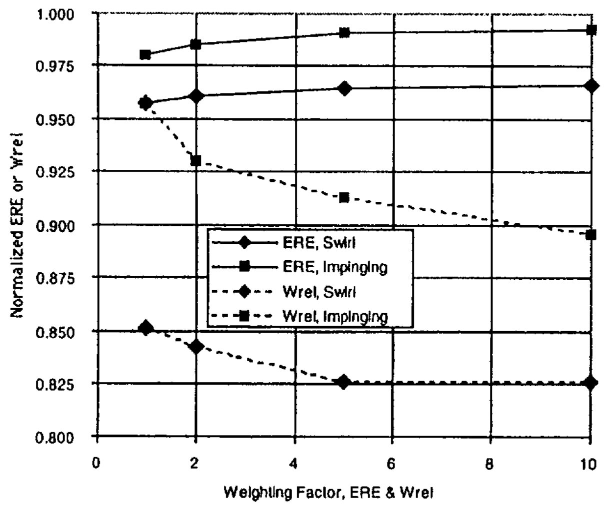


Fig. 26. Performance and weight trends for swirl and impinging elements.

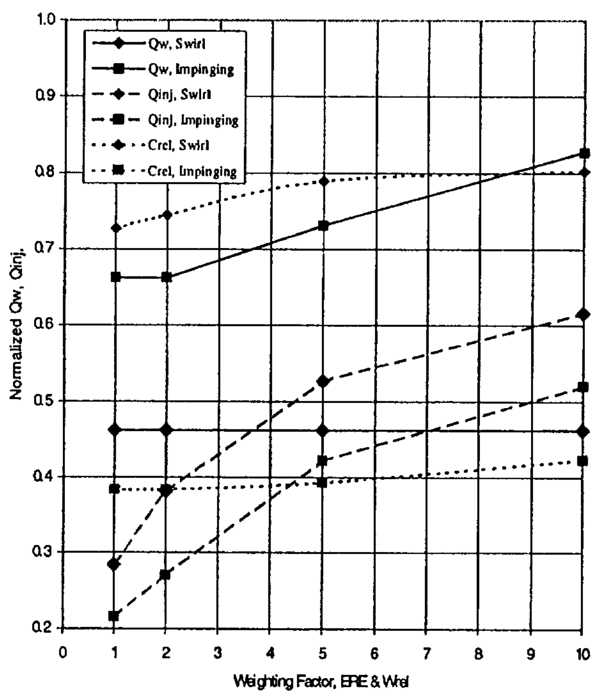


Fig. 27. Heat flux and cost trends for swirl and impinging elements.

by over 1%. As noted earlier, increasing Θ leads to increased injector heat flux. For this case, emphasis on thrust and weight tends to have an adverse affect on Q_{inj}. Relative cost, for the current model, is also increased significantly. Performance and weight trends for the swirl and impinging injector elements are shown in Fig. 26. Fig. 27 shows the heat flux and cost trends for the swirl and impinging injector elements.

4.4. Supersonic turbine for reusable launch vehicles

4.4.1. Polynomial-based RSM results for one-, two- and three-stage turbines

There are 28-unknown coefficients required for constructing the second-order response surface for the single-stage case, 78 for the two-stage and 136 for the three-stage case. Different starting points are tried to avoid local maximum and the optimum values of η, W and Δpay with the corresponding design parameters are determined. The results shown are comparable with the corresponding Meanline runs with the highest error of 5% for Δpay for single-stage turbine. The percentage error is increased to 13.5% for Δpay for two-stage turbine and to 14.6% for the three-stage turbine for Δpay indicating that the accuracies of the response surfaces constructed are poor for the two- and three-stage.

Papila et al. [58] have reduced the size of the parameter space by 80% in each coordinates, based on the optimal design selected in the original design space, to improve the accuracy of the response surfaces for these cases. The intention is to improve the fidelity of the response surface. With these refined designed spaces, substantial improvement of the response surface fit accuracy is observed for both cases by Papila *et al.* [58].

Based on the results obtained, the following observations can be made:

(1) To ascertain required predictive capability of the RSM, a two-level domain refinement strategy has been adopted by Papila et al. [58]. The accuracy of the predicted optimal design points based on this approach is shown to be satisfactory.

(2) According to the results obtained for Δpay-based optimization, the two-stage turbine gives the best Δpay result.

(3) As the number of stages increases, it is observed that efficiency improves while the weight increases, also. However, the improvement in efficiency cannot compensate for the penalty from higher weight.

(4) As shown in Fig. 28, the mean diameter, speed, and the exit blade area exhibit distinct trends. Specifically, the diameter decreases, speed increases, and annulus area decreases with increasing number of stages. It is interesting to observe that none of these design parameters are toward the limiting values, indicating that the optimal designs result from compromises between competing parametric trends. For such cases, a formal optimizer such as the present response surface method is very useful.

Table 21 gives a summary of the optimization results for one-, two- and three-stage turbines for Δpay-based optimization.

4.4.2. Higher-order polynomials and NN-based RSM for single-stage turbine

The generation of polynomial-based RS model and the training of the NN are done with 76 design points of the

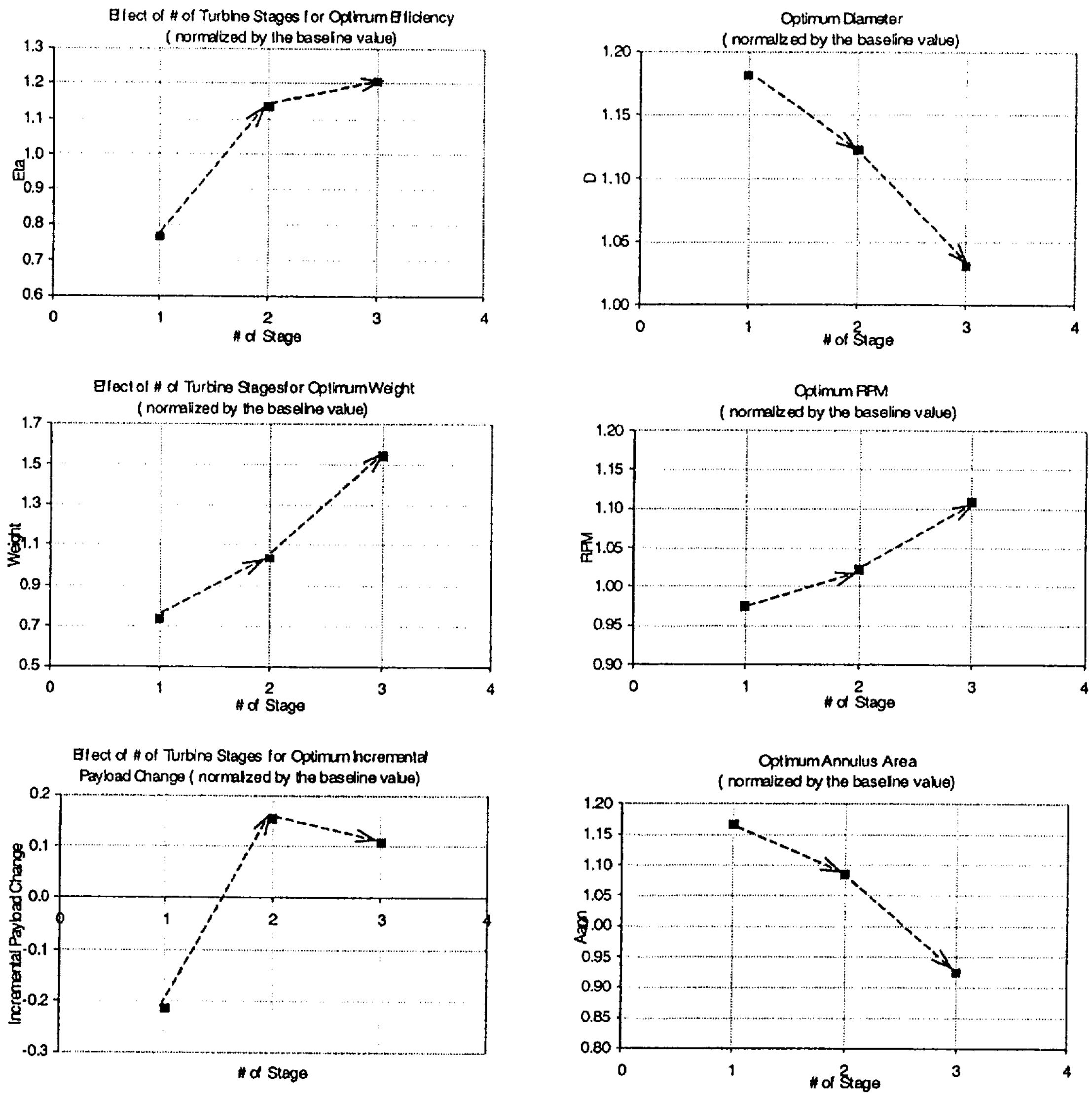


Fig. 28. Effect of the number of turbine stage on optimum design parameters; D, RPM, and A_{ann} and optimum output parameters; η, W, and Δpay calculated for Δpay-based optimization (all geometric design variables and output parameters are normalized by the baseline values).

Table 21
Optimization summary for one-, two- and three-stage turbine with response surface in original design space (all output parameters are normalized by the baseline values)

		Original design space			Refined design space		
		η_{opt}	W_{opt}	Δpay_{opt}	η_{opt}	W_{opt}	Δpay_{opt}
Δpay	1-stage	0.77	0.73	− 0.21	0.77	0.73	− 0.21
	2-stage	1.10	1.05	0.11	1.13	1.04	0.15
	3-stage	1.24	1.62	0.14	1.20	1.54	0.11

single-stage turbine. The analysis was initially done without the constraints and then with the constraints on $(AN)^2$ and V_{pitch}.

A quadratic polynomial model was initially generated. Then, cubic terms were included. Cubic terms that are products of three different variables were included

Table 22
Values of σ_a and σ for different response surfaces of η and W for the supersonic turbine

Scheme	σ_a for η (%)	σ for η (%)	σ_a for W (%)	σ for W (%)
Quadratic RS	2.51	0.90	0.82	1.27
Reduced cubic RS	1.95	1.03	0.40	1.22

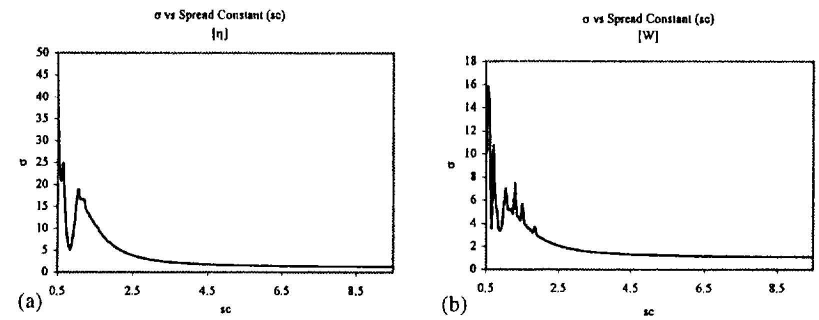


Fig. 29. Comparison of σ for different NN designed with solverbe for single-stage supersonic turbine: (a) η (%) and (b) W(lb).

because of the amount of data available and the number of levels being three. The trend of the design data also suggests the presence of some of these terms. Therefore, the initial cubic equation has 45 terms. Reduced third-order polynomial model for η and W were selected based on the relative performances of different polynomials obtained by removing terms from the initial cubic equation based on t-statistics. The cubic equation was selected based on the evaluated value of σ_a and σ. Table 22 suggests that the reduced cubic polynomial is better than the quadratic polynomial since σ_a is better for the former. The values of σ is comparable.

When constructing the NN-based response surface, the design parameters of the NN should be selected carefully since the selection of the design parameters determines the learning characteristics of the NN. For the single-stage supersonic turbine case, the variation of σ with respect to the only design parameter of solverbe network, spread constant, is plotted in Fig. 29 for both objective functions of η and W. Fig. 30 shows that for low values of spread constant, the NN has a poor performance. As the spread constant increases σ asymptotically decreases. Therefore, the appropriate spread constant is selected from the region where the performance of the network is relatively consistent. Fig. 31 shows the influence of error goal on the network performance. Unlike the case of injector (Fig. 17), a more stringent error goal for the training data does not necessarily result in better predictive capability against the test data for the single-stage turbine.

The networks designed with solverb have 37 and 75 neurons for η and W, respectively, in the hidden layer, while those designed with solverbe have 76 neurons each. The BPNN uses significantly less number of neurons by generating networks with five and 60 neurons for η and W, respectively, in a single hidden layer. The NN architectures chosen are listed in Table 23.

The accuracy of the various models is tested with the 18 additional available data and the error is shown in Table 24. Solverbe yields a relatively poor prediction for η, which might be due to over fitting, but performs well for W. Solverb is most consistent among all methods evaluated.

The optimum solutions subjected to the constraints of $(\mathrm{AN})^2$ limited to less than 1.132 (normalized with baseline value) and V_{pitch} is limited to less than 1.148 (normalized with baseline value), are presented in Table 25. Since $(\mathrm{AN})^2$ is proportional to the product of square of RPM and A_{ann}, and V_{pitch} is proportional to D times RPM, no NN or polynomial-based RSM is generated for them. By comparing the predicted optimal design by the various methods, one observes that solverbe and BPNN yield noticeably larger error in η and W, respectively. Solverb and the response surface are more consistent with both η and W. Judged by the error in predicting Δpay, it seems that the polynomial-based RSM is most accurate. However, since the real goal is to maximize Δpay, it is important to note that the actual value of Δpay for the optimal design chosen by the RSM is the worst.

From a design perspective, it is interesting to understand the impact of the constraints from A_{ann} and V_{pitch} on the optimal turbine parameters. Such an assessment is offered in Figs. 32 and 33. As D, RPM and A_{ann} decrease, η, W, V_{pitch}, AN^2 and Δpay decrease.

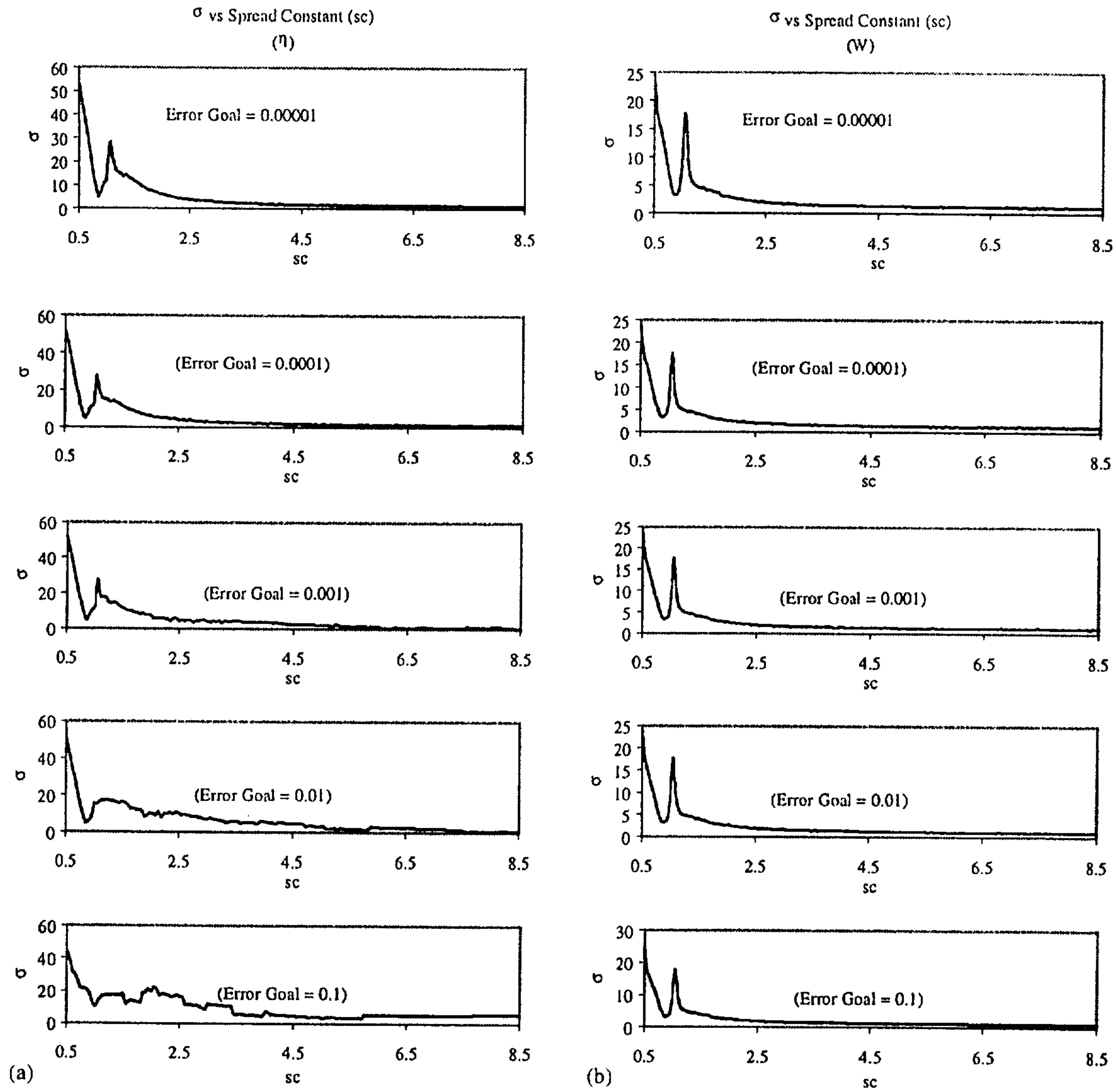


Fig. 30. Comparison of σ for different NN designed with solverb for single-stage supersonic turbine: (a) η (%) and (b) W(lb).

C_b and C_v are almost constant over the design space and they do not have any noticeable effect on the objective functions and constraints. In the case of C_v, the BPNN shows a small perturbation for the analysis with the constraint. This might be due to the mapping of some noise by BPNN. Otherwise it is unaffected by the inclusion of the constraints. The optimum stage reaction, *sr*, is zero implying that the optimum design should be that of an impulse turbine.

4.4.3. Orthogonal arrays for two-stage turbine

Although the majority of the work is based on the face centered composite design approach (FCCD), orthogonal arrays (OA) are constructed by Papila et al. [58] to investigate the efficiency of orthogonal array designs in representing the design space for two-stage turbine. A set of 249 design points is selected using orthogonal arrays. Table 26 shows the comparison of the quality of the second-order response surfaces generated for η, W and Δpay by using 1990-data generated by face centered composite design and 249-data selected by orthogonal array method.

The above table illustrates that the fidelity of the response surface generated for design space of 249 data, based on orthogonal arrays, are comparable with that of 1990 data based on the face centered criterion. The response surface models are also assessed by using 78-test data to determine the predictive accuracy of these models. Table 27 presents that the testing adjusted rms-errors of response surfaces generated are 1.65% for η and 0.96% for W using 249-data, and 1.67% for η and 1.21% for W using 1990-data.

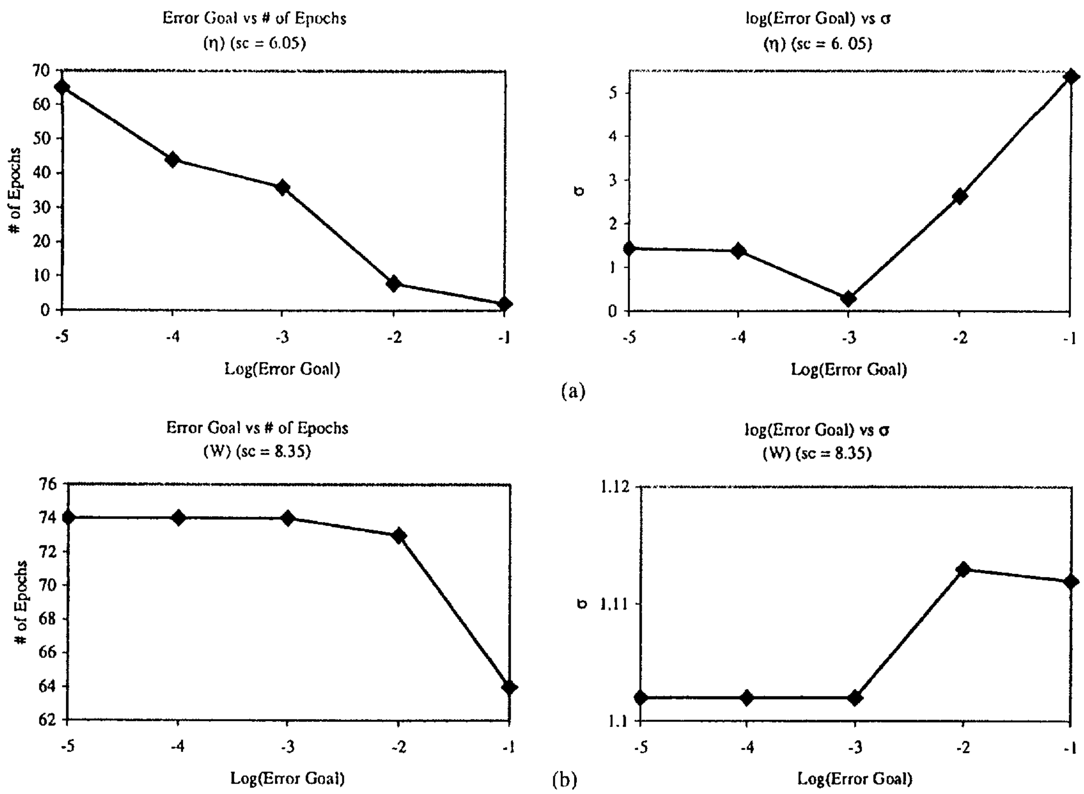


Fig. 31. Comparison of error goal versus number of epochs and σ for networks trained with solverb for single-stage supersonic turbine: (a) η (%) and (b) W(lb).

Table 23
Neural network architectures used to design the models for η and W of the supersonic turbine (sc = spread constant)

Scheme	# of layers	# of neurons in the hidden layer		# of neurons in the output layer		Error goal aimed for during training	
		η	W	η	W	η	W
RBNN (Solverbe)	2	76	76	1	1	0.0 {sc = 9.50}	0.0 {sc = 9.45}
RBNN (Solverb)	2	37	75	1	1	0.001 {sc = 6.50}	0.001 {sc = 8.35}
BPNN	2	5	60	1	1	0.001	0.001

Table 24
rms-error in predicting the values of η and W by various schemes for the supersonic turbine

Scheme	σ for η (%)	σ for W (%)
RBNN (Solverbe)	1.25	1.10
RBNN (Solverb)	0.29	1.10
BPNN	0.78	2.56
Reduced cubic RS	1.03	1.22

When these results are compared with the results of 1990-data and it is observed that the optimum η, W and Δpay are largely consistent. However, it is also observed from Fig. 34 which shows the comparison of the design variables for optimization based on (Δpay), some of the design variables are different even though optimum η, W and Δpay are consistent. This shows that there are multiple points in the design space, which yield comparable performance. Nevertheless, it remains true that the two-stage turbine is most suitable from a payload point of view.

Table 25
Optimal solutions with constraints on V_{pitch} and $(AN)^2$ for a supersonic turbine (error given in parenthesis for each prediction is in %) ($V_{pitch} = 1.148$ and $(AN)^2 = 1.132$ in all the cases) (all the variables are normalized by their respective baseline values)

Scheme	D (in)	RPM	A_{ann} (in^2)	C_v (in)	C_b (in)	sr (%)	η	W (lb)	Δpay (lb)
RBNN (Solverbe)	0.972	1.181	0.811	1.443	0.836	0.0	0.810 (5.80)	0.636 (0.74)	− 0.139 (29.80)
Meanline	0.972	1.181	0.811	1.443	0.836	0.0	**0.766**	**0.641**	**− 0.197**
RBNN (Solverb)	0.999	1.149	0.857	1.483	0.792	0.0	0.785 (1.75)	0.653 (0.17)	− 0.177 (9.16)
Meanline	0.999	1.149	0.857	1.483	0.792	0.0	**0.772**	**0.654**	**− 0.194**
BPNN	1.024	1.121	0.901	1.168	1.143	0.0	0.793 (2.49)	0.608 (8.63)	− 0.153 (21.49)
Meanline	1.024	1.121	0.901	1.168	1.143	0.0	**0.772**	**0.666**	**− 0.195**
Reduced cubic RS	0.903	1.272	0.700	1.706	0.871	0.0	0.758 (1.50)	0.591 (2.10)	− 0.194 (8.40)
Meanline	0.903	1.272	0.700	1.706	0.871	0.0	**0.746**	**0.604**	**− 0.211**

4.4.4. NN-based RSM for two-stage turbine

In order to find the optimum RBNN design for the design of the two-stage turbine design, the effect of the spread constant (sc) on the network training error is determined. Figs. 35 and 36 show the variation of solverbe network error, σ, with respect to spread constant for the NN designed for FCCD and OA data. The optimum spread constant is determined as 3.2 for 1990-training data (FCCD) from Fig. 35 and 4.3 for 249-data (OA) from Fig. 36. In spite of the fact that the spread constants are larger than 3, the training rms-errors (σ_a) are less than 0.1% for all networks designed for refined space with 249-data as shown in Fig. 37. Based on this observation, sc = 4.3 value is used for these cases for consistency.

After constructing the NN-based response surface, the NN model is tested by using 78-test data selected along the main diagonal of the design space to determine the predictive accuracy of these models. Table 28 presents the prediction rms-errors (σ) of second-order polynomial response surfaces, which are 1.65% for η and 0.96% for W using 249-data, and 1.67% for η and 1.21% for W using 1990-data. Table 28 also presents that the prediction rms-errors of response surfaces generated by solverbe RBNN are 1.36% for η and 1.30% for W, and 2.26% for η and 1.56% for W using 249-data.

Fig. 38 summarizes fitting/training and testing results of the RBNN and polynomial-based Δpay approximations for the two-stage turbine. The efficiency of the multi-level RSM approach can be observed by comparing the original and refined design space plots. From these plots, it is also possible to observe that more accurate training is possible with RBNN but testing or prediction accuracies of the RBNN and polynomial-based approximations are quite comparable.

4.5. Turbulent planar diffuser

4.5.1. Polynomial fits

Based on the D-optimal set of 35 design points selected, the 21 regressors of a full quadratic polynomial were fitted resulting in a moderate R_a^2-value of 0.810. A backward elimination of regressor terms subsequently led to the removal of five terms and an increase of R_a^2 to 0.848. The lower values of R_a^2, in comparison to the two-design-variable case, reflect the increased difficulties in obtaining a good fit when moving to higher-dimensional response surfaces. Data on the backward elimination steps are given in Table 29, which apart from R^2 and R_a^2 holds the minimum t-statistic and the number of uncertain terms with $|t_0| < 2.0$ remaining in the model. From the t-statistics information, it appears that the backward elimination improved the accuracy of remaining terms.

The next step performed was to investigate whether the 35 applied observations included outliers. A common (but not necessarily true) assumption, which enables the statistical treatment of observations, is that errors are independently and identically distributed according to a normal distribution with mean zero and variance.

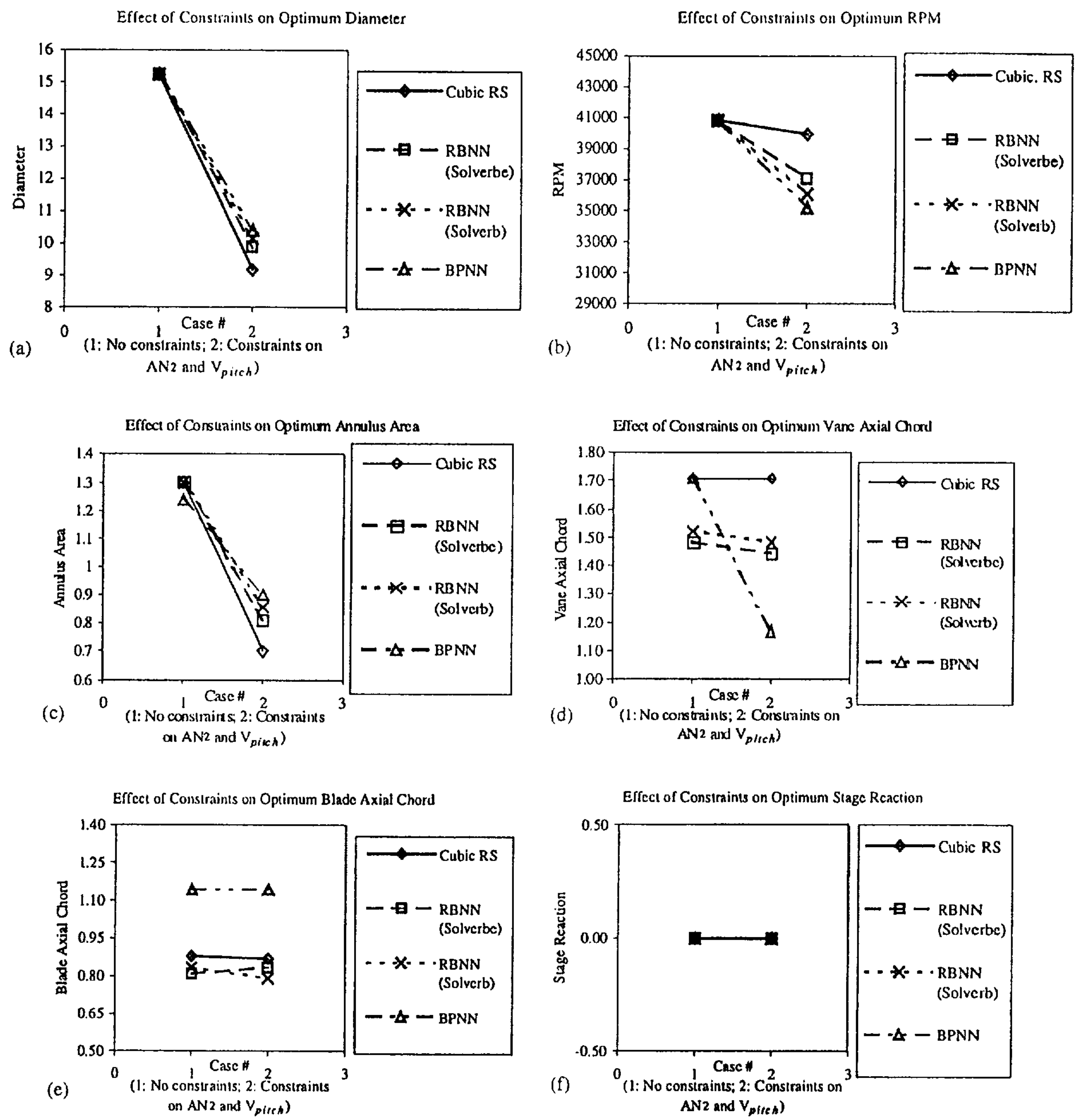


Fig. 32. Effect due to presence (case 1) or lack of constraints (case 2) on design variables: (a) optimum diameter, D (in), (b) optimum RPM, (c) optimum annulus area, A_{ann} (in^2), (d) optimum vane axial chord, C_V (in), (e) optimum blade axial chord, C_b (in), (f) optimum stage reaction, sr (%).

Thus, the distribution of response surface errors was plotted and compared to a normal distribution, with which it is expected to correspond well. From the histogram plot of the error distribution, see Fig. 39, it did not seem that there are any outliers. Four arbitrary points away from sampling points were picked to test the prediction accuracy of the polynomial-based RSM. Table 30 compares CFD-results and polynomial approximations with and without backward elimination of terms.

Again, the predictions of the response surface appear reliable, except at the last control point. This point is, however, in the non-monotonic region, so that the approximation relies on an extrapolation, which was never intended. The reduced approximation model comes closer to the CFD-results for two out of the three meaningful test points.

4.5.2. Numerical noise

While noisy data from laboratory experiments is a generally accepted fact, the presence of noise in numerical simulations seems much less recognized. Due to the complex numerical modeling techniques of CFD, the exact

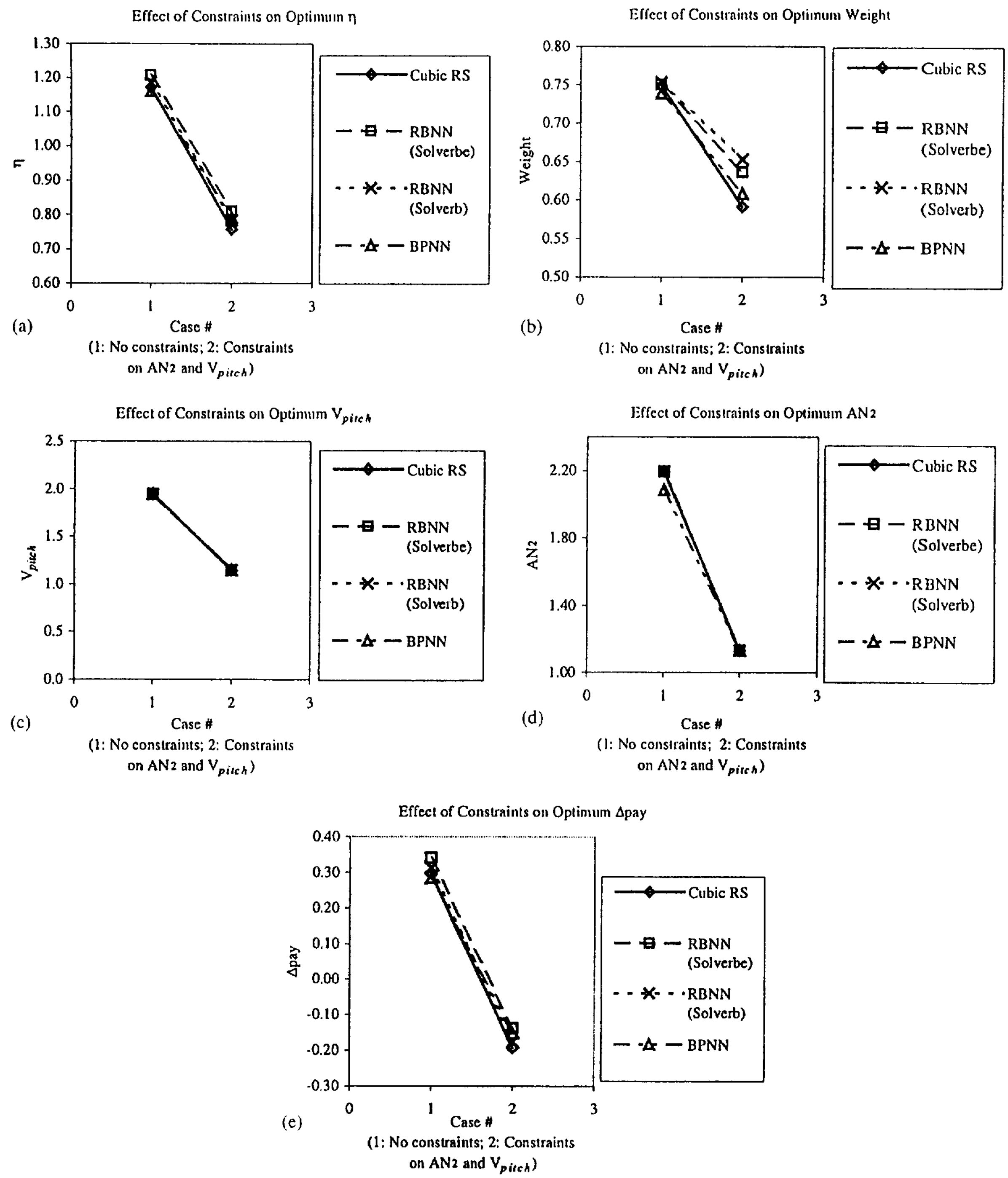


Fig. 33. Effect due to presence (case 1) or lack of constraints (case 2) on objective functions: (a) optimum efficiency, η, (b) optimum weight, W (lb), (c) optimum pitch speed, V_{pitch} (in/s), (d) optimum ($A_{\text{ann}} \times \text{RPM}$), $(\text{AN})^2$ ($\text{in}^2 \times \text{rpm}^2$), (e) optimum incremental payload, Δpay (lb).

origins of noisy responses may be difficult to pinpoint, but factors such as turbulence models, incomplete convergence, and the discretization itself are certainly influential. Here, the presence of numerical noise has been investigated. The problem of non-smooth or noisy objective functions has previously been addressed by Giunta et al. [60], who found RS approximations-based optimization to perform very robustly under such circumstances, especially when point selection is based on design of experiment techniques, such as D-optimal designs.

Table 26
The quality of the second-order response surface obtained for η, W and Δpay of two-stage turbine for 1990-data (face centered criterion) and 249-data (orthogonal arrays) (Mean values of η, W and Δpay are normalized by the baseline values)

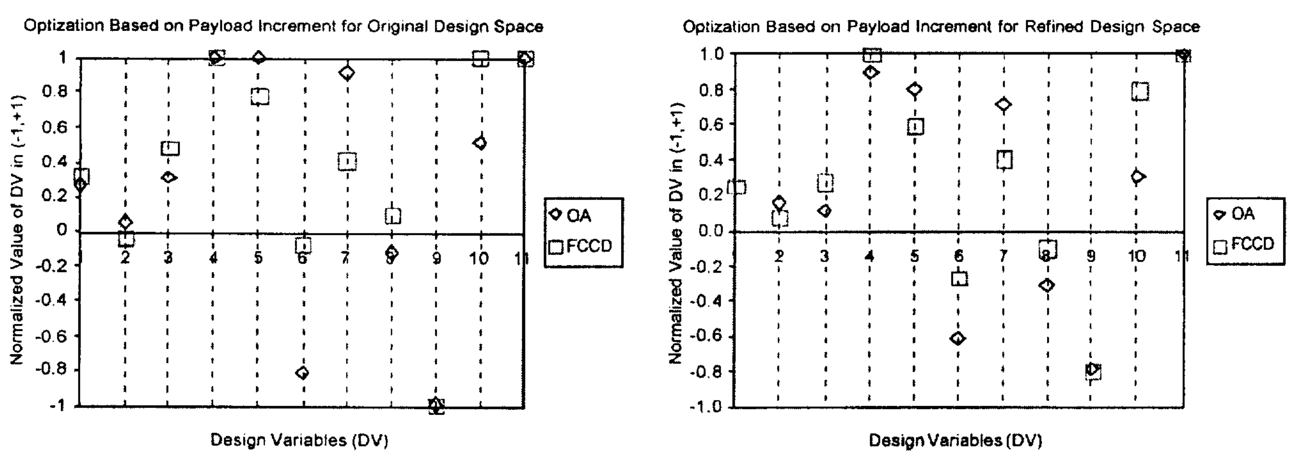
		η	W	Δpay
	R^2	0.995	0.996	0.995
	R_a^2	0.994	0.996	0.995
1990-data	rms-error	1.31%	2.56%	9.58%
	Mean	0.78	0.86	− 0.24
	R^2	0.995	0.998	0.994
	R_a^2	0.992	0.997	0.992
249-data	rms-error	2.128%	0.826%	20.68%
	Mean	0.89	0.92	− 0.11

Table 27
Testing of the second-order response surface obtained for η and W of two-stage turbine for 1990 data (FCCD criterion) and 249-data (OA) with 78-test data

# of design points	# of test data	σ for η (%)	σ for W (%)
249	78	1.65	0.96
1990	78	1.67	1.21

Limitations of the software used were felt during the application of a wall-shape parameterization in the investigation of noise. A B-spline curve with two free control points was used. Again, it was observed that the objective function oscillated due to numerical noise, but the amplitude was small. To make the noise more apparent, it was therefore necessary to refine the subdivision of the discretized line and reduce its length to 20% of the initial, so that the line spans from (0.3,0.6) to (0.302,0.602). This yielded the noisy response patterns shown in Fig. 40. The two topmost curves in this figure were determined using a relatively tight convergence criterion, and two different convection schemes—a standard first-order upwind differencing scheme (UDS) and a second-order upwind differencing scheme (SUDS). The use of different differencing schemes was carried out to estimate whether numerical diffusion does significantly dampen the generation of noise. As discussed in [59], two different CFD codes were adopted, and one seems less forgiving, in the sense that it predicts a stronger tendency for flow separation. This could possibly be explained by factors such as numerical diffusion, boundary treatments, and momentum interpolation methods adopted in the two codes.

As expected, switching to a more dissipative differencing scheme (lower order accuracy) yields a smoother response. To further illustrate this issue, one more design line curve is shown in Fig. 40, which arose from using a relatively loose, yet still reasonable, convergence criterion (using SUDS). The applied convergence criterion considers summed and normalized (by inlet flux) residuals over the entire mesh, with termination of computations once the maximum is below a certain small value ε. The loose convergence criterion in Fig. 40 was $\varepsilon = 10^{-3}$, whereas the tight tolerance was $\varepsilon = 10^{-5}$. For comparison, a convergence limit of $\varepsilon = 10^{-4}$ was applied in the CFD analyses used for response surface modeling. The overall conclusion is that the presence of some numerical noise in CFD-results is practically inevitable, although



(a) Original Design Space (b) Refined Design Space

Fig. 34. Comparison of the design variables for optimization based on payload increment (Δpay) using 1990-data (FCCD) and 249-data (OA) for both original design space and refined design space ($DV\#1$: D, $DV\#2$: RPM, $DV\#3$: A_{ann}, $DV\#4$: h_1, $DV\#5$: c_{v1}, $DV\#6$: c_{v2}, $DV\#7$: c_{b1}, $DV\#8$: c_{b2}, $DV\#9$: sr_1, $DV\#10$: sr_2, and $DV\#11$: w_{f1}). Both designs are satisfactory, demonstrating that there exist multiple optimum designs.

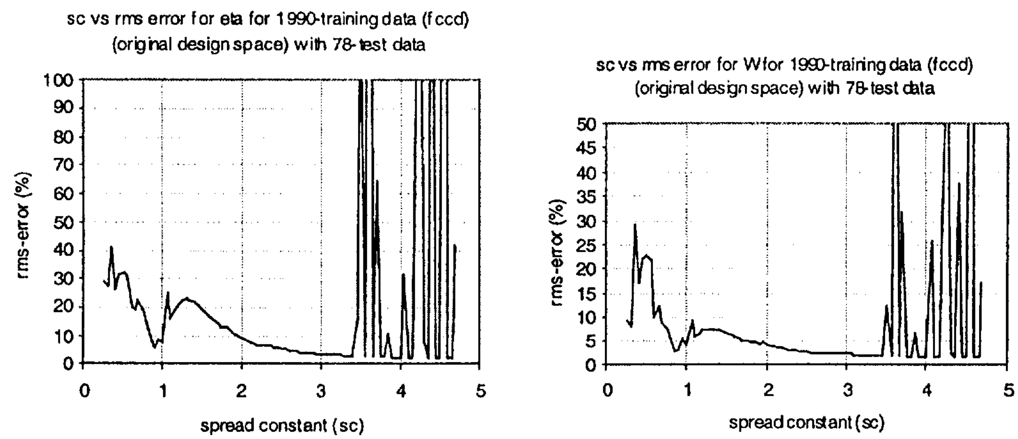


Fig. 35. Effect of spread constant (sc) on training rms-error (σ_a) of (a) η and (b) W for preliminary design of two-stage turbine for original design space using 249-training data for solverbe RBNN.

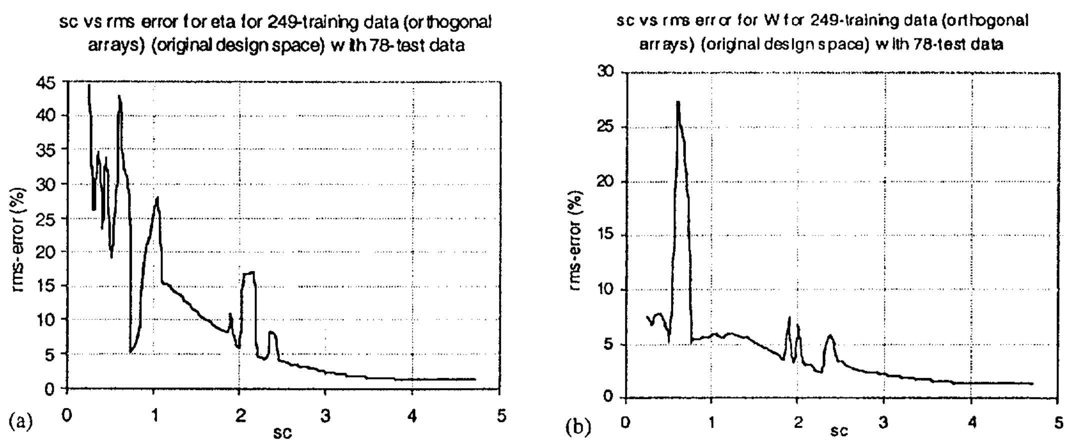
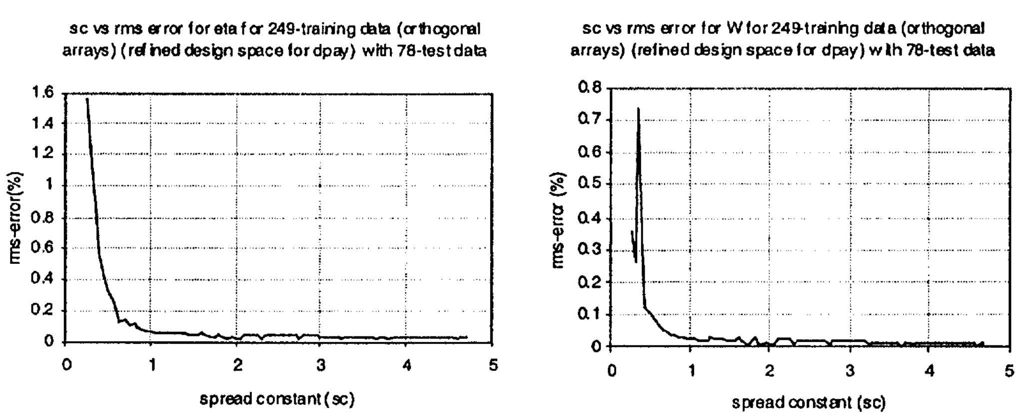


Fig. 36. Effect of spread constant (sc) on training rms-error (σ_a) of (a) η and (b) W for preliminary design of two-stage turbine for original design space using 249-training data (OA) for solverbe original RBNN.



(a) for η (Δpay optimization) (b) for W (Δpay optimization)

Fig. 37. Effect of spread constant (sc) on training rms-error (σ_a) of for preliminary design of two-stage turbine for refined design space using 249-training data (OA) for solverbe, RBNN.

Table 28
Testing the RBNN and second-order polynomial response surface obtained for η and W for preliminary design of two-stage turbine (original design space)

Number of training data	Number of test data	sc	σ for η (%) using RBNN (Solverbe)	σ for η (%) using polynomial-based	σ for W (%) using RBNN (Solverbe)	σ for W (%) using polynomial-based
249	78	4.3	1.365	1.648	1.305	0.959
1990	78	3.2	2.263	1.672	1.557	1.214

its magnitude depends on choice of code and modeling techniques. Here, a technique such as polynomial-based RSM can be effective in smoothing out the undesirable fluctuations.

4.5.3. Optimum diffuser designs

In the optimum design using B-spline parameterization, both the monotonicity constraint and four out of five side constraints are active. As already mentioned, the response surface constructed to guarantee wall monotonicity becomes too restrictive. To compensate for this, a one-dimensional search in the direction of the steepest gradient was conducted starting at the optimum design point estimated by RSM:

$$y = y^* + \alpha \nabla F. \tag{31}$$

This search is terminated as soon as designs turn non-monotonic, yielding a new optimum point at the edge of the true feasible domain and an increase in the optimum pressure recovery coefficient from 0.7208 to 0.7235. Fig. 41 compares the optimum wall contours determined by RSM using B-splines and polynomial shapes. The optimum B-spline shape compares well to the optimum polynomial one, so it is not surprising that there is no significant gain compared to this case. The largest differences in shape are found in the later part of the expansion, where the shape has less impact on the overall performance, as separation is small in either case. Thus, the close resemblance of optimum inlet shapes is reassuring in terms of the credibility of the optimization algorithm. A CFD-analysis of the five-design-variable optimum design yields a pressure recovery coefficient of 0.7193, a little below the predicted value, as in the two-design-variable case. The improvement from the two design variable case (0.7185–0.7193) indicates that there is not much potential for further gains. Furthermore, for comparison, Fig. 41 also contains the corresponding wall contour determined using search optimization techniques. The optimum wall shape found by search optimization can be described as truly bell-shaped, without a "plateau" similar to the one found in the results of RSM-optimization. There appears to be a distinct difference in optimum shapes from the two different optimization approaches, which must be ascribed to the combination of optimization accentuating modeling differences and a relatively small scatter in diffuser performances.

Fig. 42 highlights the use of a response surface approximation for the optimum shape of a two-dimensional diffuser. As illustrated, within the fidelity of the analysis tool, there are often multiple design points that meet the design objectives. It is interesting to note that different diffuser shapes are found to yield essentially the same performance. The response surface model is ideally suitable for such situations.

4.6. Low Reynolds number wing model

4.6.1. Polynomial fits

For the 3-D wing case, the response is the flight power index, $C_L^{3/2}/C_D$, and the design space consists of design variables maximum camber, y_c, and wing aspect ratio, AR. Quadratic, cubic and quartic order polynomials are tested for the best approximations for data sets containing 9, 15 and 25 simulated data points (see Table 34). The predicted rms-errors are calculated for each of the model and are shown in Table 31. As shown in this table, Model 4 gives the smallest predicted rms-error for the cases involving 9 and 15 simulated data points, whereas, Model 12 allows the smallest predicted rms-error for the case involving 25 simulated data points.

4.6.2. Comparison of radial-basis and back-propagation networks

The predictive accuracy of neural networks depends not only on the training data but also on the parameters used to define the network. The best values for these parameters cannot be determined by using only training data, because typically one can obtain very small errors for the training data with a wide range of these parameters. However, the performance of NN can be examined using test data.

For the radial-basis network, one important issue is to investigate the magnitude of error in the test data to help to select the spread constant. For the back-propagation network, where cost of computation is an issue, the effect of number of neurons on the cost and accuracy should be checked. It was noticed that for the back-propagation

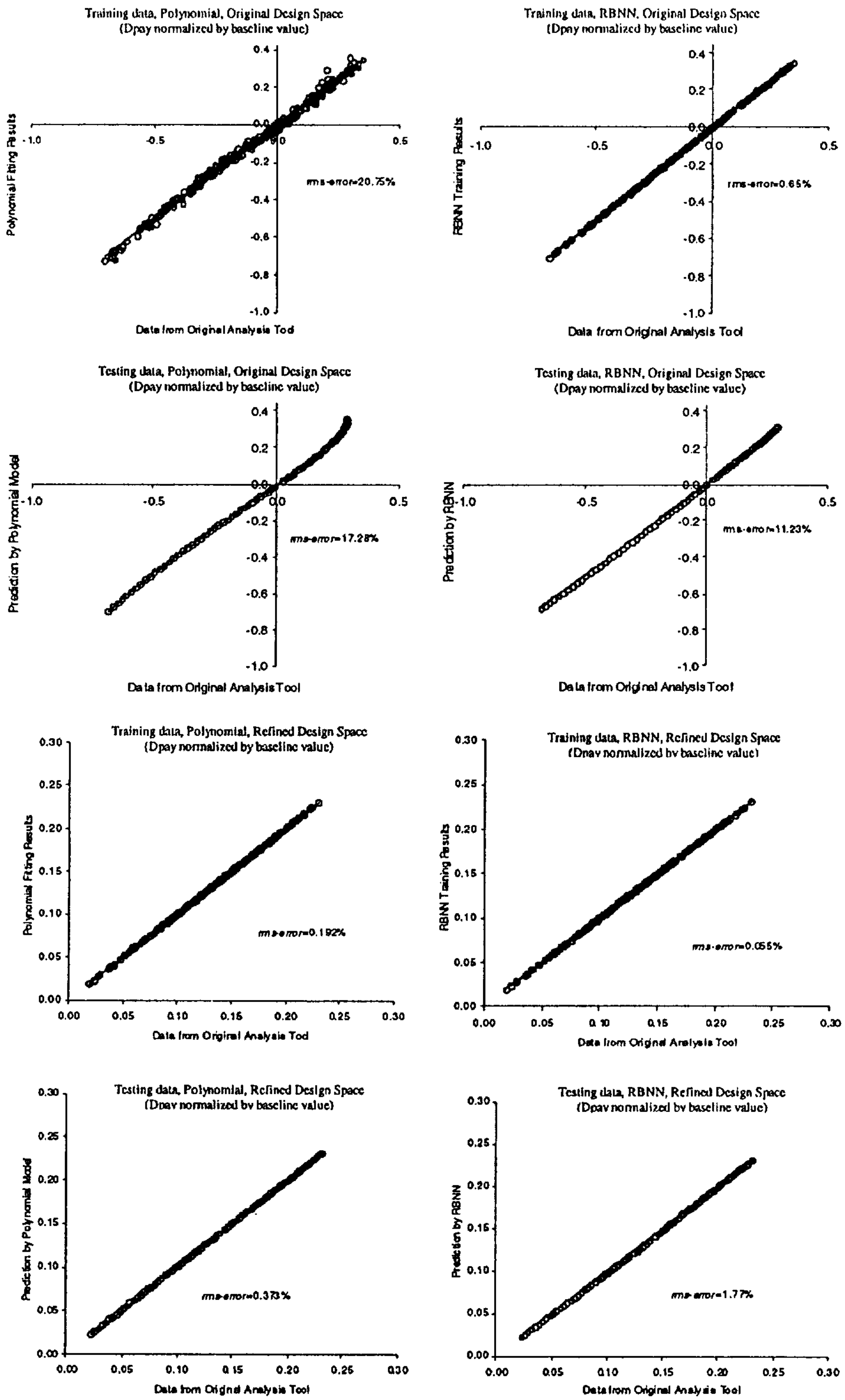


Fig. 38. Comparison of NN and polynomial-based representations for two-stage supersonic turbine. Plotted are the original and predicted values of Δpay. A perfect fit will result in a 45° line. The training and testing data are selected based on the orthogonal arrays with the D-optimal criterion. There are 11 design variables, 249 training data (OA), and 78 testing data in both original and refined design spaces (the values for Δpay are normalized by the baseline value).

Table 29
Backward elimination procedure for polynomial-based RSM in five variables

Terms	Min $\|t_0\|$	No. $\|t_0\| < 2.0$	R^2	R_a^2	Comments
21	0.05	15	0.922	0.811	
20	0.23	14	0.922	0.823	Removed y_3^2
19	0.45	12	0.922	0.834	Removed y_4^2
18	0.53	9	0.921	0.841	Removed $y_1 y_4$
17	0.97	8	0.919	0.848	Removed $y_2 y_5$
*16	1.22	6	0.915	0.848	Removed y_2^2
15	1.57	5	0.909	0.844	Removed $y_1 y_3$

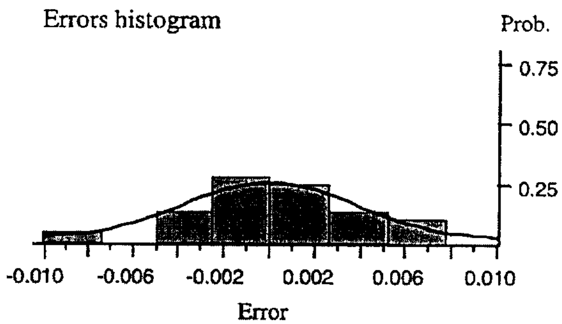


Fig. 39. Distribution of response surface errors at sampling points and the corresponding normal distribution curve (same mean and variance).

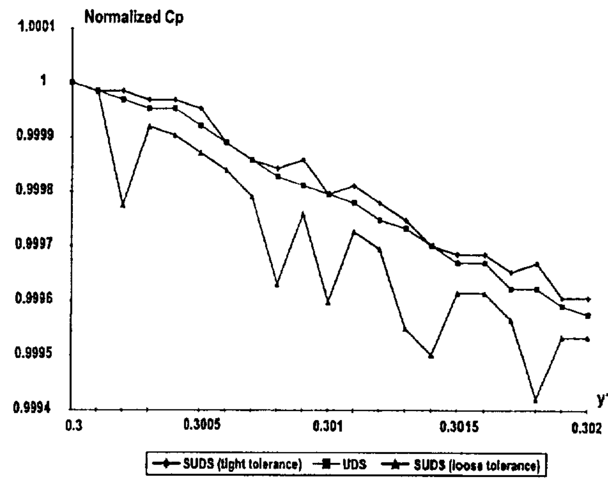


Fig. 40. Normalized C_p-values along a straight line in the design space. The results are for two different differencing schemes and two different residual levels used as convergence criterion. (UDS: upwind differencing scheme and SUDS: second-order upwind differencing scheme.)

network, using four neurons gave a good compromise of accuracy and cost. For the radial-basis network, it was found that the error and the number of iteration required for convergence are extremely sensitive to the value of spread constant. After extensive experimentation, the spread constant was chosen as 1.175.

For the 3-D wing case, both radial-basis NN and back-propagation networks are applied. In order to be able to make comparisons between these networks, the training time histories are summarized in Tables 32 and 33. These tables show that both are efficient in the training of 9-, 15- and 25-simulation training data sets. However, as the data size increases, the back-propagation network exhibits a growth rate in terms of the number of epochs, indicating that it is more CPU time intensive for larger data sizes. As far as accuracy is concerned, both networks perform well exhibiting improved predictive capabilities as the number of training points increases from 9- to 25-simulation for y_c interpolations (Fig. 43). For this case, both methods reproduced the original 9-simulation accurately but both failed to predict accurately the interpolation points at $y_c = 0.0125$, 0.025, 0.075 and 0.0875 with the rms-error of the test data of

Table 30
Comparison between CFD-solutions and polynomial-based RSM-predictions

Y_1	Y_2	Y_3	Y_4	Y_5	F	$\hat{F}$ (full)	$\hat{F}$ (reduced)
0.5	0.5	0.5	0.5	0.5	0.7171	0.7148	0.7126
1.0	0.5	0.0	0.5	1.0	0.7174	0.7210	0.7174
0.25	0.75	0.25	0.75	0.25	0.7148	0.7185	0.7162
0.0	0.5	1.0	0.5	0.0	0.6943	0.7333	0.7283

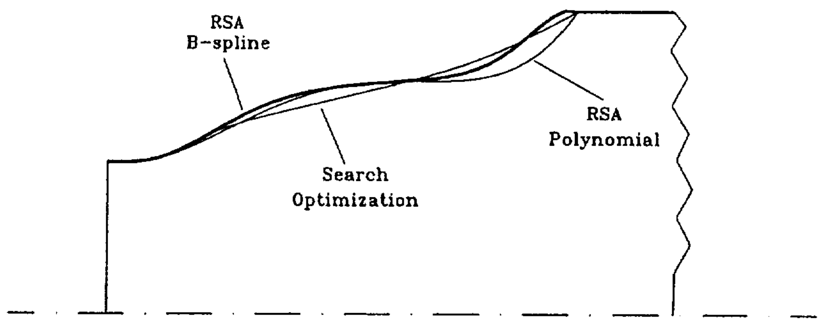


Fig. 41. Comparison of optimum wall shapes using polynomial and B-spline representations, respectively.

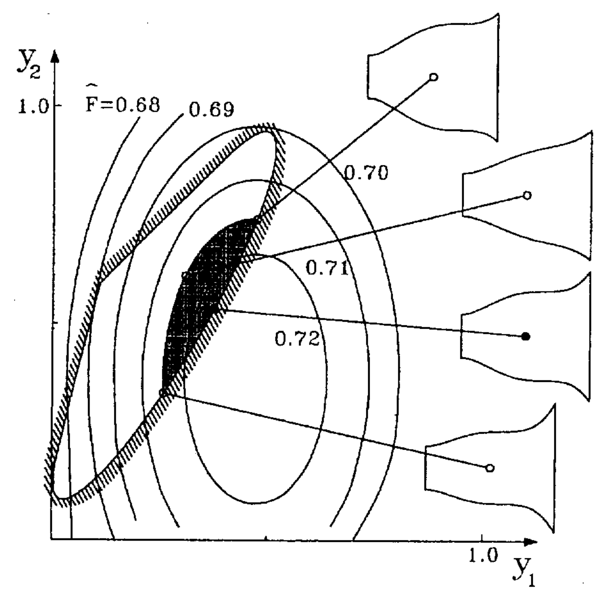


Fig. 42. Contour plot of response surface for diffuser design. Solid circle indicates the optimal region. The hatched part of the feasible space comprises designs with performance within 1% of the optimal. Corresponding shapes are indicated to the right. The results indicate that multiple design points meet the goal.

1.68 for back-propagation network and 1.04 for radial-basis network (Fig. 43a). Fig. 43b shows that adding 6 new points at AR = 2 and 4 at y_c = 0, 0.05 and 0.1 (15-simulation training data set) does not significantly improve the 6 interpolated values (rms-error values of 1.369 for back-propagation network and 1.029 for radial-basis network). However, with the addition of 10 new points at y_c = 0.025 and 0.075 at AR = 1, 2, 3, 4 and 5 (25-simulation training data set) both the back-propagation network and the radial-basis network can accurately capture the overall behavior of the aerodynamic data as shown in Fig. 43c. The rms-error now is 0.141 for back-propagation network and 0.106 for radial-basis network. For AR interpolations, the back-propagation network resulted in lower rms-error values when compared to the rms-error values of radial-basis networks (Fig. 44). For the 9-point simulation training data, the rms-error of radial-basis network (rms-error = 11.12) is quite high when compared to the rms-error of back-propagation (rms-error = 1.172) (Fig. 44a). For this case, adding 6 new points at AR = 2 and 4 at y_c = 0, 0.05 and 0.1 significantly improves the rms-error value for radial-basis (rms-error = 0.87) as shown in Fig. 44b. With the addition of 10 new points to 15-simulation data at y_c = 0.025 and 0.075 at AR = 1, 2, 3, 4 and 5, the rms-error decreases further to 0.7 for radial-basis networks, and 0.026 for back-propagation (Fig. 44c). The results indicate that the back-propagation network is quite accurate for small to modest number of data for the cases investigated and it is also more consistent than that of the radial-basis network. However, as indicated in Tables 32 and 33. In terms of computing time or epochs, back-propagation network scales unfavorably with respect to the number of data used. In other words, the back-propagation network is competitive for modest data size while the radial-basis network is more effective for larger data size. More information will be presented when the 2-D airfoil case that involves substantially larger data size is discussed.

4.6.3. Comparison of radial-basis neural network and polynomial-based techniques

For the 3-D wing model, the outputs of the solverb radial-basis NN, along with the results of the polynomial-based technique, are compared for different size of the data. It must be noted that the network parameters used to obtain radial-basis network results are sc = 1.175 and error goal = 10^{-2}. Fig. 45 illustrates the comparison between the NN and polynomial-based outputs based on the 9-simulation training data set. For this case, both methods reproduced the original 9-simulation accurately but both failed to predict accurately the interpolation points at y_c = 0.025 and 0.075 with rms-errors at the test data of 1.04 for both the NN and polynomial-based methods. Furthermore, it is seen that the error estimate of 1.116 of Table 31 is a gross underestimate. Note that by the time there are 25 data points, Table 31 predicts an

Table 31
Predicted rms-error, σ, for different polynomial models for 3-D wing model: 9-, 15-, and 25-simulation data sets (the shaded models indicate the best fit)

Model no.	Model	σ for 9 data	σ for 15 data	σ for 25 data
1	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6$	0.8047	0.5172	0.7800
2	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7\mathrm{AR}^3$	0.8047	0.5475	0.8007
3	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3$	0.8047	0.5172	0.5524
4	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7\mathrm{AR}y_C^2$	0.1162	0.0738	0.6590
5	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{AR}y_C^2$	—	—	0.3207
6	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{AR}y_C^2 + c_9y_C\mathrm{AR}^2$	—	—	0.3262
7	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{AR}y_C^2 + c_9\mathrm{AR}^3$	—	—	0.6961
8	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{AR}y_C^2 + c_9y_C\mathrm{AR}^2 + c_{10}\mathrm{AR}^3$	—	—	0.3350
9	$c_1\mathrm{AR} + c_2\mathrm{AR}y_C + c_3y_C + c_4y_C^2 + c_5 + c_6y_C^3 + c_7\mathrm{AR}y_C^2$	—	—	0.4248
10	$c_1\mathrm{AR} + c_2\mathrm{AR}y_C + c_3y_C + c_4y_C^2 + c_5 + c_6\mathrm{AR}^2y_C$	—	—	0.8044
11	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{Ar}y_C^2 + c_9y_C^4$	—	—	0.2383
12	$c_1\mathrm{AR}^2 + c_2\mathrm{AR} + c_3\mathrm{AR}y_C + c_4y_C + c_5y_C^2 + c_6 + c_7y_C^3 + c_8\mathrm{Ar}y_C^2 + c_9y_C^4 + c_{10}\mathrm{AR}y_C^3$	—	—	0.1073

Table 32
Training history of radial-basis networks with Solverb for 3-D wing model

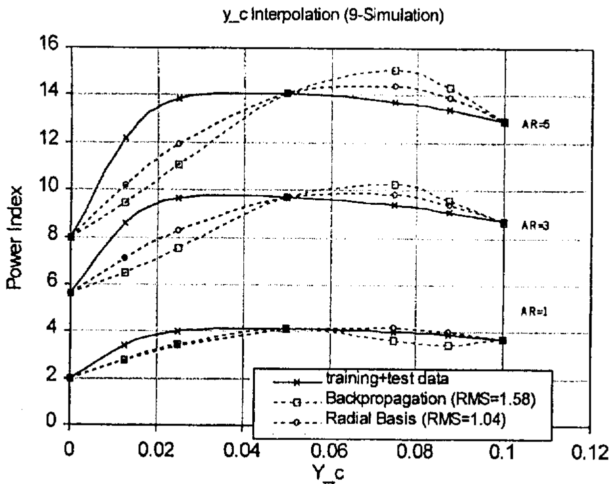
NN No.	# of simulations	# of neurons	# of epochs	Steady-state error	Spread constant	Error goal
1	9	8	7	10^{-16}	1.175	10^{-2}
2	15	12	11	10^{-4}	1.175	10^{-2}
3	25	20	19	10^{-3}	1.175	10^{-2}

Table 33
Training history of back-propagation networks with Trainlm for 3-D wing model

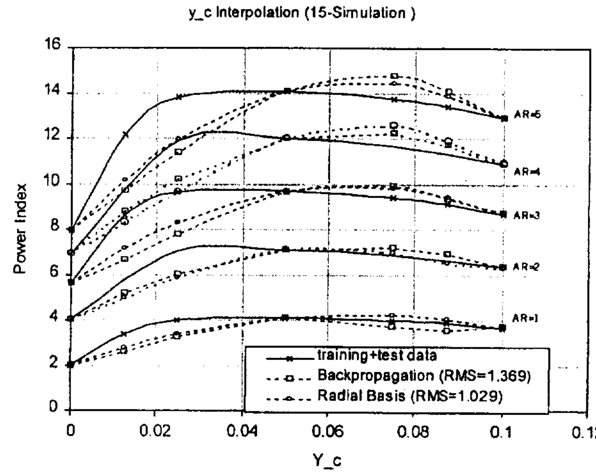
NN No.	# of simulations	# of neurons	# of epochs	Steady-state error	Error goal
1	9	4	23	4.5×10^{-4}	10^{-2}
2	15	4	12	8.5×10^{-3}	10^{-2}
3	25	4	105	9.96×10^{-3}	10^{-2}

error of 0.659. The reason for this problem is that rms-error estimates are not reliable when the number of coefficients is close to the number of points (7 versus 9 for this case). In addition, these estimates assume random noise and that underlying function is quadratic. Fig. 45b shows that adding 6 new points $\mathrm{AR} = 2$ and 4 at $y_c = 0, 0.05$ and 0.1 does not help noticeably to improve the 6 interpolated values (rms-error values of 1.029 for both). However, with the addition of 10 new points at $y_c = 0.025$ and 0.0075 at $\mathrm{AR} = 1, 2, 3, 4$ and 5 (25-simulation training data set) both the NN and polynomial-based techniques accurately capture the overall behavior of the aerodynamic data as shown in Fig. 45c. The generalization of the NN with 25-simulation is further assessed by comparing additional interpolated values at different y_c and AR at $y_c = 0.0125$ and 0.0875 at $\mathrm{AR} = 1, 2, 3, 4$ and 5. The rms-errors now are 0.142 for the polynomial and 0.221 for the NN, which are more in the line with the prediction in Table 31.

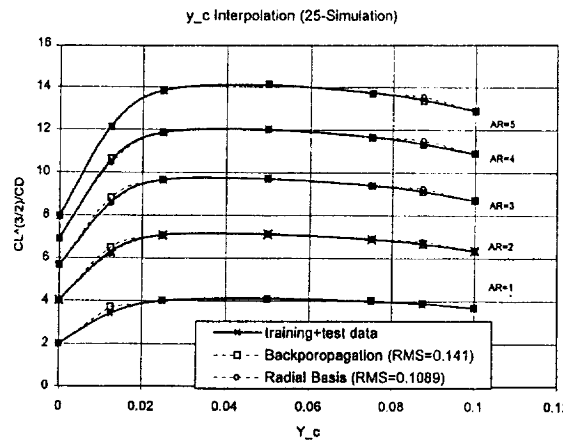
These comparisons illustrate that both neural network and conventional polynomial fitting methods do a good job as the number of points is increased.



(a) 9-Simulation Training



(b) 15-Simulation Training

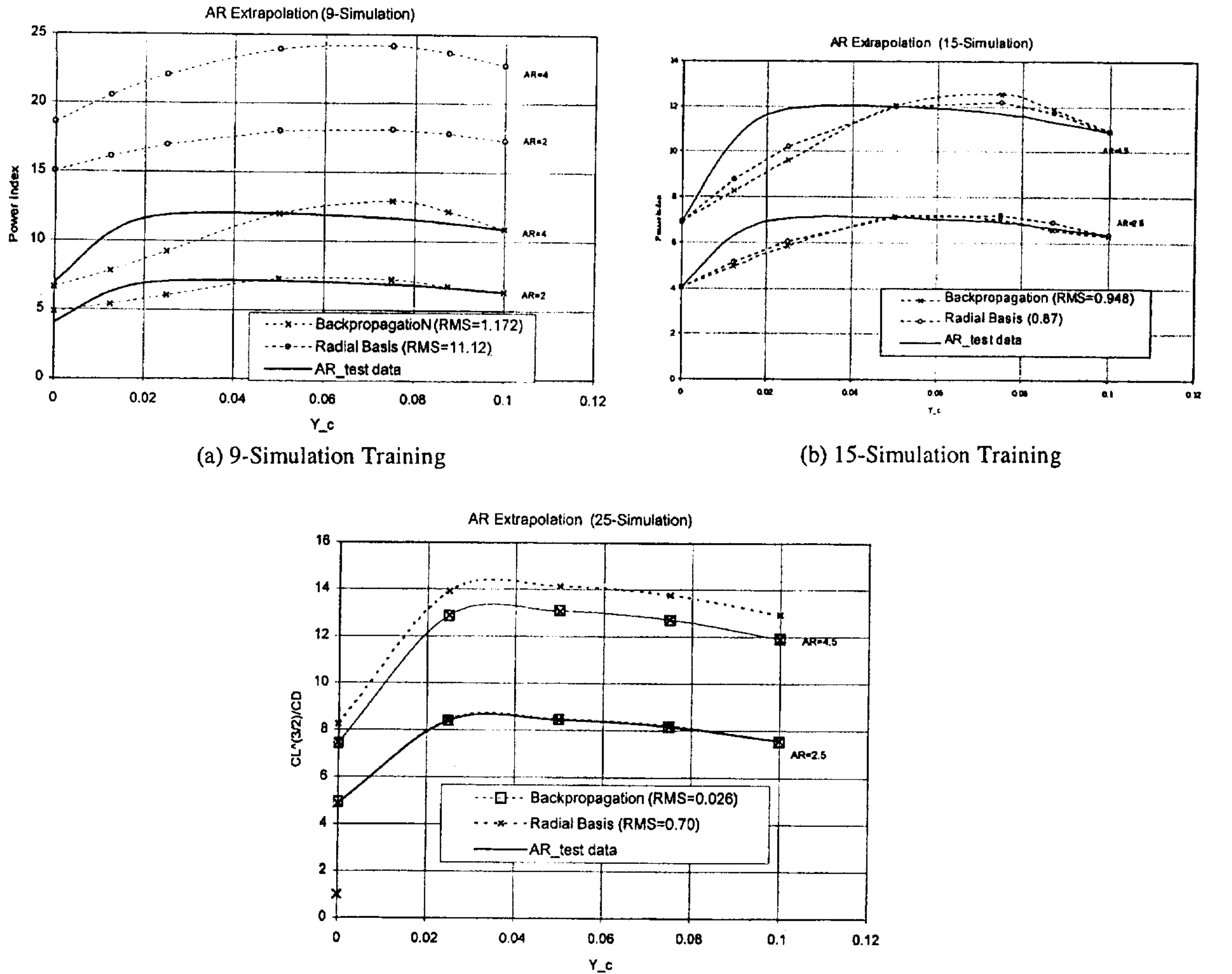


(c) 25-Simulation Training

Fig. 43. Comparison of radial basis network with back-propagation network results for 3-D wing model (for y_c interpolation) (design parameters: sc = 1.175, error goal = 0.1 for radial-basis, and #of neurons = 4, error goal = 0.01 for back-propagation).

5. Conclusion and future directions

Recent experiences in utilizing a global optimization methodology, based on polynomial and neural network techniques, for aerodynamics and rocket propulsion components are summarized. Global optimization methods can utilize the information collected from various sources and by different tools. These methods offer multi-criterion optimization, handle the existence of multiple design points and trade-offs via insight into the entire design space, can easily perform tasks in parallel, and are often effective in filtering the noise intrinsic to numerical and experimental data. Another advantage is that these methods do not need to calculate the sensitivity of each design variable locally. The global optimization method can be particularly effective with either a polynomial-based response surface or a neural network when information from different computational, experimental and analytical sources needs to be assembled. In this article, we present recent experiences in utilizing the global optimization methodology for tasks related to the preliminary design of a supersonic turbine, multi-criterion design of three different types of injector element (shear co-axial, impinging, and swirl co-axial), performance of a low Reynolds number wing, and shape optimization of a turbulent flow diffuser. A successful optimal design technique often needs to address the issues related to the selection of appropriate training data for constructing the global model, employment of the statistical and testing tools to identify appropriate global models, existence of multiple design selections and related trade-offs, and consideration of noises intrinsic to numerical and experimental data. These issues are discussed. It is seen that the global optimization method can naturally take the confidence level of the data into account, offers a number of designs with comparable performance, and



(a) 9-Simulation Training

(b) 15-Simulation Training

(c) 25-Simulation Training

Fig. 44. Comparison of radial basis network with back-propagation network results for 3-D wing model (for AR interpolation) (design parameters: sc = 1.175, error goal = 0.1 for radial-basis, and # of neurons = 4, error goal = 0.01 for back-propagation).

allows designers to make a more informed decision. We have reviewed direct evidences that demonstrate that appropriate selection of design points can significantly reduce the number of data required for constructing the global model. In particular, while the FCCD approach can be effective with modest number of design variables, OA with D-optimal selection criterion seems to be effective when the number of design variables becomes higher. Regarding the relative merits between polynomials and neural networks, based on the results reviewed, we can make the following summary:

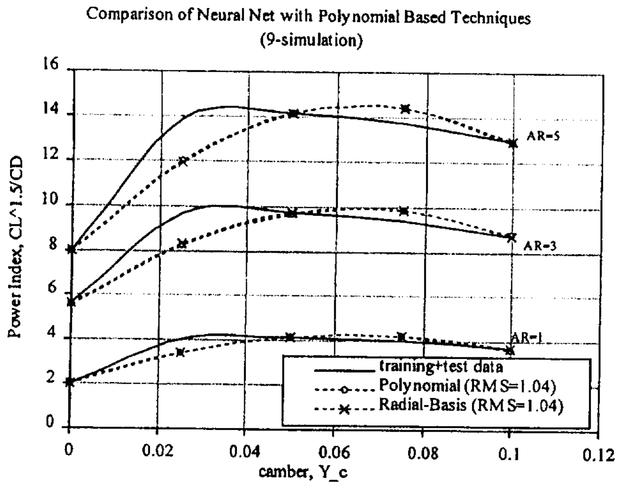
(1) Higher-order polynomials usually perform better than lower-order polynomials as they have more flexibility. However, exceptions have been noticed which demands that appropriate statistical measures be taken to determine the best terms to include in an expression.

(2) Both NN and polynomial-based RSM can perform comparably for modest data sizes.

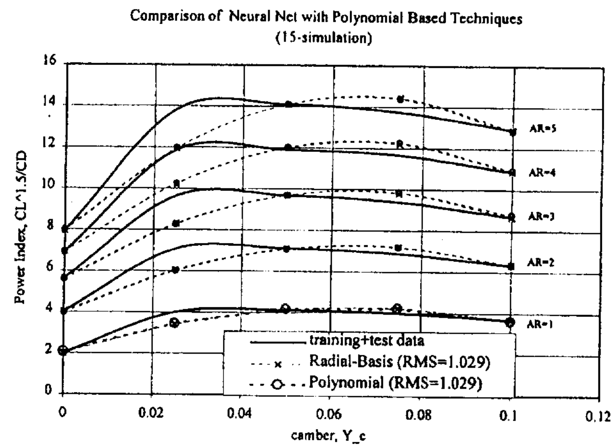
(3) Among all the NN configurations, RBNN designed with solverb seems to be more consistent in performance.

(4) Radial basis networks, even when designed efficiently with solverb, tend to have many more neurons than a comparable back-propagation with tan- or log-sigmoid neurons in the hidden layer. The basic reason for this is the fact that the sigmoid neurons can have outputs over a large region of the input space, while radial basis neurons only respond to relatively small regions of the input space. Thus, larger input spaces require more radial basis neurons for training.

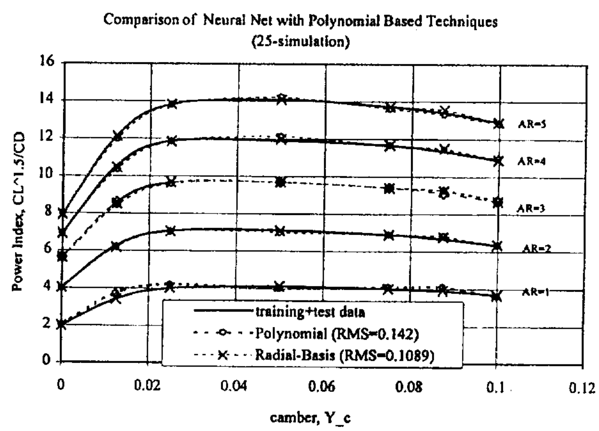
(5) Configuring a radial basis network often takes less time than that for a back-propagation network because the training process for the former is linear in nature.



(a) 9-Simulation Training



(b) 15-Simulation Training



(c) 25-Simulation Training

Fig. 45. Comparison of radial basis NN results with polynomials for 3-D wing model (design parameters: sc = 1.175, error goal = 0.1 for radial-basis networks).

(6) While the transfer function employed by any neural network is non-linear in general, the RBNN, with the combined feature of flexibility and linear regression is more accurate than BPNN, which requires solution of non-linear systems.

(7) The comparisons demonstrate that for this case there are no significant differences between the NN- and polynomial-based RSM. The results of polynomial-based methods, though, suggest that when the error is mostly due to modeling rather than noise, the error estimates of the polynomial-based technique can be substantially off.

(8) The NN technique has shown the potential of fitting the data much better than the polynomial-based technique. However, this was achieved by using the test data to select the parameters like spread constant of the NN which appear to greatly affect the predictive accuracy. That is, it was not possible to use only the training data to select the best set of parameters. This indicates that because the NNs do not provide the statistical information given by polynomial-based methods, using both test data and training data is very important in designing the network.

(9) With the large number of points, and the high-order polynomial, the statistical predictions of the polynomial-based results matched very well the error at the test data.

(10) The neural networks, when trained appropriately, can be used to generate additional data to enhance the data set for constructing polynomials. Such a combined approach has been demonstrated in [38] for injector design.

(11) The criteria for selecting the database exhibit significant impact on the efficiency and effectiveness of the

construction of the response surface. For example, effectiveness of using OA to select the database is demonstrated by Papila et al. [58].

(12) A multi-level approach can be applied to identify the optimal design points with substantially higher accuracy.

There are a number of outstanding issues that need to be addressed. In the following, we list several topics that we consider important for future research.

(1) *Is it possible to develop a comprehensive technique by combining NN and polynomial-based RS techniques to help reduce the required data size for optimization?*

Specifically, the work done by Rai and Madavan [27–29], Madavan et al. [22], and Shyy et al. [38] suggests that NN can be effectively used to supplement the existing training data to help to generate a more accurate polynomial. The RBNN may lack satisfactory filtering properties in some cases [37,39]. However, once trained, RBNN can generate additional design data to feed the polynomial-based RSM. Polynomials possess the intrinsic filtering capability. The evaluation of the nature of the fluctuations from the data generated by RBNN, and the investigation into whether polynomials can use the data effectively, is planned. These features have been addressed in this article.

(2) *What are the keys to develop a more robust and flexible NN configuration?*

This has been a topic of research for a long period of time. In this article, a review is presented to address the issues related to the training characteristics of the different networks used. There are other important issues, which needs to be addressed in future research. For example, the possibility of using a more versatile RBNN in terms of a variable design parameter, unlike the current situation where the variable has the same value throughout the domain, should be addressed. Objective means to determine the NNs performance via statistical tools, especially for RBNN since it employs a linear model to determine the weight associated with each neuron needs to be investigated.

(3) *What is the scaling rule between the number of neurons, and computing time, versus number of input/output variables and the size of the design data?*

There are several rules of thumb for BPNN in the literature (e.g., [10,11,61]) but little information exists for RBNN.

(4) *How can one address the need for generating training and testing data most economically and effectively?*

The effect of the selection of the design points on accuracy, scaling and performance of polynomial-based RSM has been addressed. The same has yet to be done for NN.

Acknowledgements

The present work is partially supported by NASA Marshall Space Flight Center.

Appendix A. Training data

The results of training data are given in Tables 34–41.

Table 34
Training data sets for 3-D wing model

Training data set # 1 (9-simulation)			Training data set # 2 (15-simulation)			Training data set # 3 (25-simulation)		
AR	y_c	$C_L^{3/2}/C_D$	AR	y_c	$C_L^{3/2}/C_D$	AR	y_c	$C_L^{3/2}/C_D$
1	0	2.0011	1	0	2.0011	1	0.0	2.0011
1	0.05	4.1224	1	0.05	4.1224	1	0.025	4
1	0.1	3.6865	1	0.1	3.6866	1	0.05	4.1224
3	0	5.6398	2	0	4.03	1	0.075	3.99
3	0.05	9.6873	2	0.05	7.12	1	0.1	3.6866
3	0.1	8.6806	2	0.1	6.34	2	0.0	4.03
5	0	7.9413	3	0	5.6398	2	0.025	7.07
5	0.05	14.0942	3	0.05	9.6873	2	0.05	7.12
5	0.1	12.8951	3	0.1	8.6806	2	0.075	6.89
			4	0	6.92	2	0.1	6.34
			4	0.05	11.99	3	0.0	5.6398
			4	0.1	10.87	3	0.025	9.64
			5	0	7.9414	3	0.05	9.6873
			5	0.05	14.0942	3	0.075	9.39
			5	0.1	12.8951	3	0.1	8.6806
						4	0.0	6.92
						4	0.025	11.86
						4	0.05	11.99
						4	0.075	11.66
						4	0.1	10.87
						5	0.0	7.9414
						5	0.025	13.83
						5	0.05	14.0942
						5	0.075	13.73
						5	0.1	12.8951

Table 35
Test data sets for 3-D wing model based on AR and y_c

Test set # 1 for y_c		Test set # 2 for y_c		Test set # 3 for y_c		Test set # 1 for AR		Test set # 2 for AR	
AR	y_c	AR	y_c	AR	y_c	AR	y_c	AR	y_c
1	0.025	1	0.025	1	0.0125	2	0	2.5	0
1	0.075	1	0.075	1	0.0875	2	0.025	2.5	0.025
3	0.025	2	0.025	2	0.0125	2	0.05	2.5	0.05
3	0.075	2	0.075	2	0.0875	2	0.075	2.5	0.075
5	0.025	3	0.025	3	0.0125	2	0.1	2.5	0.1
5	0.075	3	0.075	3	0.0875	4	0	4.5	0
		4	0.025	4	0.0125	4	0.025	4.5	0.025
		4	0.075	4	0.0875	4	0.05	4.5	0.05
		5	0.025	5	0.0125	4	0.075	4.5	0.075
		5	0.075	5	0.0875	4	0.1	4.5	0.1

Table 36
Performance and heat flux responses for $O/F = 4$ for the shear co-axial injector element (Tables 36–38 together contain 45 data points used as the training set)

O/F	V_f/V_o	L_{comb} (in)	ERE (%)	Q (Btu/in^2 s)
4.0	4.0	4.0	92.9	0.753
4.0	4.0	5.0	96.0	0.753
4.0	4.0	6.0	97.6	0.753
4.0	4.0	7.0	98.6	0.753
4.0	4.0	8.0	99.0	0.753
4.0	6.0	4.0	95.0	0.928
4.0	6.0	5.0	97.1	0.928
4.0	6.0	6.0	98.5	0.928
4.0	6.0	7.0	99.2	0.928
4.0	6.0	8.0	99.4	0.928
4.0	8.0	4.0	96.6	1.10
4.0	8.0	5.0	98.2	1.10
4.0	8.0	6.0	99.1	1.10
4.0	8.0	7.0	99.4	1.10
4.0	8.0	8.0	99.6	1.10

Table 37
Performance and heat flux responses for $O/F = 6$ for the shear co-axial injector element

O/F	V_f/V_o	L_{comb} (in)	ERE (%)	Q (Btu/in^2 s)
6.0	4.0	4.0	92.9	0.691
6.0	4.0	5.0	96.0	0.691
6.0	4.0	6.0	97.6	0.691
6.0	4.0	7.0	98.6	0.691
6.0	4.0	8.0	99.0	0.691
6.0	6.0	4.0	95.0	0.642
6.0	6.0	5.0	97.1	0.642
6.0	6.0	6.0	98.5	0.642
6.0	6.0	7.0	99.2	0.642
6.0	6.0	8.0	99.4	0.642
6.0	8.0	4.0	96.6	0.741
6.0	8.0	5.0	98.2	0.741
6.0	8.0	6.0	99.1	0.741
6.0	8.0	7.0	99.4	0.741
6.0	8.0	8.0	99.6	0.741

Table 38
Performance and heat flux responses for $O/F = 8$ for the shear co-axial injector element

O/F	V_f/V_o	L_{comb} (in)	ERE (%)	Q (Btu/in^2 s)
8.0	4.0	4.0	92.9	0.588
8.0	4.0	5.0	96.0	0.588
8.0	4.0	6.0	97.6	0.588
8.0	4.0	7.0	98.6	0.588
8.0	4.0	8.0	99.0	0.588
8.0	6.0	4.0	95.0	0.512
8.0	6.0	5.0	97.1	0.512
8.0	6.0	6.0	98.5	0.512
8.0	6.0	7.0	99.2	0.512
8.0	6.0	8.0	99.4	0.512
8.0	8.0	4.0	96.6	0.493
8.0	8.0	5.0	98.2	0.493
8.0	8.0	6.0	99.1	0.493
8.0	8.0	7.0	99.4	0.493
8.0	8.0	8.0	99.6	0.493

Table 39
Data used to test the polynomials and NN for the shear co-axial injector element (the table contains 20 data points used as the testing set)

O/F	V_f/V_o	L_{comb} (in)	ERE (%)	Q (Btu/in^2 s)
4.0	5.0	4.0	94.4	0.812
4.0	5.0	5.0	96.9	0.812
4.0	5.0	6.0	98.1	0.812
4.0	5.0	7.0	99.1	0.812
4.0	5.0	8.0	99.4	0.812
4.0	7.0	4.0	96.0	1.014
4.0	7.0	5.0	97.9	1.014
4.0	7.0	6.0	98.8	1.014
4.0	7.0	7.0	99.4	1.014
4.0	7.0	8.0	99.6	1.014
6.0	5.0	4.0	94.4	0.642
6.0	5.0	5.0	96.9	0.642
6.0	5.0	6.0	98.1	0.642
6.0	5.0	7.0	99.1	0.642
6.0	5.0	8.0	99.4	0.642
6.0	7.0	4.0	96.0	0.691
6.0	7.0	5.0	97.9	0.691
6.0	7.0	6.0	98.8	0.691
6.0	7.0	7.0	99.4	0.691
6.0	7.0	8.0	99.6	0.691

Table 40
Propellant momentum ratio as a function of propellant pressure drops: shear co-axial injector element

ΔP_f	ΔP_o 200	180	160	150	140	120	100
200	**1.49**	1.42	1.33	**1.30**	1.25	1.16	**1.06**
180	1.57	1.50	1.41	1.37	1.32	1.22	1.11
160	1.67	1.59	1.50	1.45	1.40	1.30	1.18
150	**1.73**	1.64	1.54	**1.49**	1.44	1.34	**1.22**
140	1.79	1.70	1.60	1.55	1.50	1.39	1.27
120	1.93	1.83	1.72	1.67	1.61	1.50	1.37
100	**2.11**	2.00	1.89	**1.83**	1.77	1.64	**1.49**

Table 41
Design data for a shear co-axial injector element with ΔP_o and $\Delta P_f = 200$ psi

ΔP_o	ΔP_f	L_{comb}	α	ERE	Q_w	$H_{impinge}$	W_{rel}	C_{rel}
200	200	2	15	NA	0.85	0.84	0.923	1.083
200	200	2	20	85	0.85	0.62	0.923	1.083
200	200	2	30	92.8	0.85	0.39	0.923	1.083
200	200	2	45	95.4	0.85	0.23	0.923	1.083
200	200	2	50	95.8	0.85	0.19	0.923	1.083
200	200	4	15	91	0.85	0.84	1	1.083
200	200	4	20	95.2	0.85	0.62	1	1.083
200	200	4	30	96.8	0.85	0.39	1	1.083
200	200	4	45	98.1	0.85	0.23	1	1.083
200	200	4	50	98.4	0.85	0.19	1	1.083
200	200	6	15	95.6	0.85	0.84	1.077	1.083
200	200	6	20	97.8	0.85	0.62	1.077	1.083
200	200	6	30	98.5	0.85	0.39	1.077	1.083
200	200	6	45	99.2	0.85	0.23	1.077	1.083
200	200	6	50	99.4	0.85	0.19	1.077	1.083
200	200	8	15	98.3	0.85	0.84	1.154	1.083
200	200	8	20	99.1	0.85	0.62	1.154	1.083
200	200	8	30	99.4	0.85	0.39	1.154	1.083
200	200	8	45	99.6	0.85	0.23	1.154	1.083
200	200	8	50	99.7	0.85	0.19	1.154	1.083

References

[1] Baysal O, Eleshaky ME. Aerodynamic design optimization using sensitivity analysis and CFD. AIAA J 1992;30:718–25.

[2] Lambert PA, Lecordix JL, Braibant V. Constrained optimization of nacelle shapes in Euler flow using semi-analytical sensitivity analysis. Struct Optim 1995;10:239–46.

[3] Reuther J, Alonso JJ, Rimlinger MJ, Jameson A. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. Comput fluids 1999;28:675–700.

[4] Elbanna HM, Carlson LA. Aerodynamic sensitivity coefficients using the three-dimensional full potential equations. J Aircraft 1994;31:1071–7.

[5] Dadone A, Valorani M, Grossman B. Smoothed sensitivity equation method for fluid dynamic design problems. AIAA J 2000;38:418–26.

[6] Sobieszczanski-Sobieski J, Haftka RT. Multidisciplinary aerospace design optimization: survey of recent developments. Struct Optim 1997;14:1–23.

[7] Shyy W, Papila N, Tucker PK, Vaidyanathan R, Griffin L. Global design optimization for fluid machinery applications. The Second International Symposium On Fluid Machinery And Fluid Engineering, October, Beijing, China, 2000.

[8] Bauer Jr. KW, Parnell GS, Myers DA. Response surface methodology as a sensitivity analysis tool in decision analysis. J Multi-Criteria Decision Anal 1999;8:162–80.

[9] Madsen JI. Design optimization of internal flow devices. Philosophy of doctorate thesis, Aalborg University, 1998.

[10] Greenman RM. 2-D high-lift aerodynamic optimization using neural networks. NASA TM-1998-112233, June 1998. p. 146.

[11] Carpenter WC, Barthelemy J-FM. A comparison of polynomial approximations and artificial neural nets as response surface. Struct optim 1993;5:166–74.

[12] Faller WE, Schereck SJ. Unsteady fluid mechanics applications of neural networks. 33rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 95-0529, 1995.

[13] Fan X, Herbert T, Haritonidis JH. Transition control with neural networks. 33rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 95-0674, 1995.

[14] Greenman RM, Roth KR. High-lift optimization design using neural network. Proceedings of DETC98, ASME Computers in Engineering Conference, Atlanta, Georgia, DETC98/CIE-6006, 1998.

[15] Greenman RM, Roth KR. Minimizing computational data requirements for multi-element airfoils using neural networks. 37th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 99-0258, 1999.

[16] Huang SY, Miller LS, Steck JE. An exploratory application of neural networks to airfoil design. 34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 94-0501, 1994.

[17] Illi OJ, Greitzer FL, Kangas LJ, Reeve TJ. An artificial neural network system for diagnosing gas turbine engine fuel faults. 48th Meeting of the Mechanical Failures Prevention Group (MFPG48), 1994.

[18] Kangas LJ, Greitzer FL, Illi OJ. TEDANN: turbine engine diagnostic artificial neural network. The American Defense Preparedness Association (ADPA) National Symposium and Exhibition on Advanced Information Systems and Technology, 1994.

[19] Lavretsky E. Ordered neural networks: formulation, analysis, design and application. 37th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 99-0106, 1999.

[20] Lawrence S, Giles CL, Tsoi AC. What size of neural network gives optimal generalization? Convergence properties of back propagation. Technical Report, UMIACS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, University of Maryland, 1996.

[21] Lawrence S, Tsoi AC, Giles CL. Correctness, efficiency, extendibility and maintainability in neural network simulation. International Conference on Neural Networks, ICNN 96, June 2–6, Washington DC. New York: IEEE Press, 1996. p. 474–9.

[22] Madavan NK, Rai MM, Huber FW. Neural net-based redesign of transonic turbines for improved unsteady aerodynamic performance. AIAA/SAE/ASME/ASEE 35th Joint Propulsion Conference, AIAA 99-2522, June 20–24, 1999.

[23] Maghami PG, Sparks Jr DW. Design of neural networks for fast convergence and accuracy. 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Long Beach, California, AIAA 98-1780, 1998.

[24] Maghami PG, Sparks Jr DW. Design of neural networks for fast convergence and accuracy: dynamics and control. IEEE Trans Neural Networks 2000;11(1):113–23.

[25] Nikolaidis E, Long L, Ling Q. Neural networks and response surface polynomials for design of vehicle joints. 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Long Beach, California, AIAA-98-1777, 1998.

[26] Norgaard M, Jorgenson CC, Ross JC. Neural network prediction of new aircraft design coefficients. NASA TM-112197, May 1997. p. 20.

[27] Rai MM, Madavan NK. Application of artificial neural networks to the design of turbomachinery airfoils. 36th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 2000-0169, 1998.

[28] Rai MM, Madavan NK. Aerodynamic design using neural networks. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, AIAA 98-4928, 1998.

[29] Rai MM, Madavan NK. Improving the unsteady aerodynamic performance of transonic turbines using neural networks. 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 2000-0169, 2000.

[30] Ross JC, Jorgenson CC, Norgaard M. Reducing wind tunnel data requirements using neural networks. NASA TM-112193, May 1997. p. 15.

[31] Sparks Jr DW, Maghami PG. Neural networks for rapid modeling and analysis. 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Long Beach, California, AIAA 98-1779, 1998.

[32] Stepniewski SW, Jorgenson CC. Toward a more robust pruning procedure for MLP networks. NASA/TM-1998-112225, April 1998. p. 16.

[33] Stepniewski SW, Greenman RM, Jorgenson CC, Roth KR. Designing compact feedforward neural models with small training data sets. 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA 2000-0170, 2000.

[34] Orr MJL. Introduction to radial basis function networks. Centre for Cognitive Science, Edinburgh University, EH 9LW, Scothland, UK, 1996 (http://www.anc.ed.ac.uk/~mjo/rbf.html).

[35] Orr MJL. Optimising the widths of RBFs. Fifth Brazilian Symposium on Neural Networks, Belo Horizonte, Brazil, 1998.

[36] Orr MJL. Recent advances in radial basis function networks. Technical Report for Institute for Adaptive and Neural Computation Division for Informatics, Edinburgh University, EH 9LW, Scotland, UK, 1999 (http://www.anc.ed.ac.uk/~mjo/rbf.html).

[37] Papila N, Shyy W, Fitz-Coy N, Haftka RT. Assessment of neural net and polynomial-based techniques for aerodynamic applications. 17th Applied Aerodynamics Conference, Norfolk, VA, AIAA 99-3167, 1999.
[38] Shyy W, Tucker PK, Vaidyanathan R. Response surface and neural network techniques for rocket engine injector optimization. AIAA/SAE/ASME/ASEE 35th Joint Propulsion Conference, AIAA 99-2455, June 20–24, 1999.
[39] Vaidyanathan R, Papila N, Shyy W, Tucker PK, Griffin LW, Fitz-Coy N, Haftka RT. Neural network-based and response surface-based optimization strategies for rocket engine component design. Eighth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, 2000.
[40] JMP version 3. Statistics and graphics guide. SAS Institute Inc., 1998.
[41] Myers RH, Montgomery DC. Response surface methodology—process and product optimization using designed experiments. New York: Wiley, 1995.
[42] Rumelhart DE, McClelland JL, the PDP Group. Parallel distributed processing, vol. I and II. Cambridge, MA: MIT Press, 1986.
[43] Dernuth H, Beale M. Matlab neural network toolbox. The Math Works Inc, 1992.
[44] Kosko B. Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[45] Jang J-SR, Sun C-T, Mizutani E. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[46] Unal R, Braun RD, Moore AA, Lepsch RA. Response surface model building using orthogonal arrays for computer experiments. 19th Annual International Conference of the International Society of Parametric Analysis, New Orleans, Louisiana, 1997.
[47] Unal R, Lepsch RA, McMillin ML. Response surface model building and multidisciplinary optimization using D-optimal designs. Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Paper No. 98-4759, 1998.
[48] Haftka RT, Scott EP, Cruz JR. Optimization and experiments: a survey. Appl Mech Rev 1998;51(7):435–48.
[49] Papila M, Haftka RT. Uncertainty and wing structural weight approximations. 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, St Louis—MO, AIAA-99-1312, 1999.
[50] Unal R, Braun RD, Moore AA, Lepsch RA. Design optimization on cost basis using Taguchi's orthogonal arrays. Proceedings of the Annual American Society for Engineering Management (ASEM) Conference, 1996.
[51] Unal R, Dean EB. Design for cost and quality: the robust design approach. http://mijuno.larc.nasa.gov/pap/robdes/robdes.html, 1995.
[52] Unal R, Dean EB. Taguchi approach to design optimization for quality and cost: an overview. Proceedings of the International Society of Parametric Analysts 13th Annual Conference, New Orleans, Louisiana, 1991.
[53] Dean EB. Taguchi methods from the perspective of competitive advantage. http://akao.larc.nasa.gov/dfc/tm.html, 1996.
[54] Owen A. Orthogonal arrays for: computer experiments, integration and visualization. Statist Sinica 1992; 2(2):439–52.
[55] Tucker PK, Shyy W, Sloan JG, An integrated design/optimization methodology for rocket engine injectors. AIAA/SAE/ASME/ASEE 34th Joint Propulsion Conference, AIAA-98-3513, 1998.
[56] Tucker PK, Shyy W, Vaidyanathan R. An optimization-based approach to injector element design. AIAA/SAE/ASME/ASEE 35th Joint Propulsion Conference, Paper No. 2000-3220, 2000.
[57] Calhoon D, Ito J, Kors D. Investigation of gaseous propellant combustion and associated injector-chamber design guidelines. Aerojet Liquid Rocket Company, NASA Cr-121234, Contract NAS3-13379, 1973.
[58] Papila N, Shyy W, Griffin L, Huber F, Tran K. Preliminary design optimization for a supersonic turbine for rocket propulsion. 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Huntsville, AL, Paper No. 2000-3242, 2000.
[59] Madsen JI, Shyy W, Haftka RT. Response surface techniques for diffuser shape optimization. AIAA J 2000; 38:1512–18.
[60] Giunta AA, Dudley JM, Narducci R, Grossman B, Haftka RT, Mason WH, Watson LT. Noisy aerodynamic response and smooth approximations in HSCT design. AIAA Paper 94-4376, 1994.
[61] Fujita O. Statistical estimation of the number of hidden units for feedforward neural networks. Neural Networks 1998;11:851–9.