Computer Assisted Mechanics and Engineering Sciences, 14: 509–523, 2007.

Applications of improved SVM framework in modeling in mechanics

Krishnan V. Pagalthivarthi
Dept. of Applied Mechanics, Indian Institute of Technology
New Delhi 110016

Ankush Mittal
Dept. of Electronics and Comp. Engineering, Indian Institute of Technology
Roorkee 247667

Puneet Goyal, Gagan R. Gupta
Dept. of Computer Science and Engineering,, Indian Institute of Technology
New Delhi 110016

(Received in the final form December 27, 2006)

The problem of empirical data modeling is pertinent to several mechanics domains. Empirical data modeling involves a process of induction to build up a model of the system from which responses of the system can be deduced for unobserved data. Machine learning tools can model underlying non-linear function given training data without imposing prior restriction on the type of function. In this paper, we show how Support Vector Machines (SVM) can be employed to solve design problems involving optimizations over parametric space and parameter prediction problems that are recurrent in engineering domain. The problem considered is diffuser design where the optimal value of pressure recovery parameter can be obtained very efficiently by SVM based algorithm even in a large search space. In addition, locating the position of points on a string vibrating in a damped medium serves as an appropriate prediction problem. A grid-searching algorithm is proposed for automatically choosing the best parameters of SVM, thus resulting in a generic framework. The results obtained by SVM are shown to be theoretically sound and a comparison with other approaches such as spline interpolation and Neural Networks shows the superiority of our framework.

Keywords: conical diffuser, turbulent flow, string vibration, support vector machine, parameter grid searching, optimal pressure recovery, neural networks

1. Introduction

Machine Learning Tools (MLT) offer solutions to the problems that are difficult to solve using traditional mathematical decomposition techniques. The potential benefits of MLTs are the ability to learn from interaction with the environment such as in data acquisition, few restrictions on the functional relationships, and an inherent ability to generalize training information to similar situations. Previous work in the application of MLTs in fluid and solid dynamics has hinted at the promise of MLTs as function approximators, optimizers, and response surface based design tools [9, 20, 23]. Several MLTs, especially Artificial Neural Networks (ANN) and fuzzy logic have shown considerable improvement over the traditional method of modeling a system [12, 16, 22].

The ANN based paradigm involves deciding the structure of ANN, i.e., the number of hidden layers, the number of neurons, the learning algorithm, etc. In the next phase, the ANN parameters are learnt by training on complete data having several feature vectors. The trained ANN can then

respond to test data feature vector, where a particular value of a variable is to be estimated (i.e., the regression problem) or a class is to be assigned to the feature vector (i.e., classification problem).

Traditional ANN approaches suffer difficulties with generalization; producing models that can overfit the data [10]. This is a consequence of the optimization algorithms used for parameter selection and the statistical measures used to select the 'best' model. Also, ANN suffers from the curse of dimensionality and it has been suggested in the literature [11] that the only practical way to overcome it is to incorporate prior knowledge about the function to be modeled. However, this task is very difficult. In addition, selection of ANN structure crucially determines its performance, yet the structure is mostly selected by trial-and-error basis.

Recently, Support Vector Machines (SVM) developed by [24] are gaining popularity due to many attractive features, and promising empirical performance. As will be elaborated later in the paper, SVM minimizes an upper bound on the expected risk, as opposed to ANN that minimizes the error on the training data. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning. In addition, SVMs have a clear geometrical interpretation and SVM training is guaranteed to find the global minimum of the cost function. Support vector classifiers have already become competitive with the best available techniques for classification. In the recent past, its excellent performances on regression and time series prediction have been demonstrated in various studies, for instance, [19, 24].

In this paper, we show how SVM based framework can be utilized for two mechanics problems. The first one is a practical conical diffuser design problem that is of great interest to the industry and uses available design knowledge as well as experimental data amassed over the years. Diffuser design problem involves searching for the optimal value of pressure recovery over the parametric space formed by area ratio and non-dimensional diffuser length. Thus, it forms a good representative of general design problem.

The second problem pertains to general class of problems in which response needs to be predicted for vibrating systems under damped conditions. SVM is trained with data of vibrating string, which can then be employed to predict the string vibration for variable position at any time. The experimental results for both the problems show superior performance in comparison to other approaches such as Neural Networks and interpolation.

1.1. Contributions of the paper

The contributions of the paper can be summarized as follows. The paper introduces SVM to the domain of mechanics problems, where Neural Networks have been predominantly used. We show how with very less training data, SVM based framework can be utilized to give optimal design. An algorithm known as PSFOD is proposed to efficiently search for parameters in order to generate the best design. The common hurdle in using any MLT is selection of parameters and structure, which requires expertise and significant study. In order to widen the utility of SVM based framework, a grid-based searching algorithm is proposed for selection of the best SVM model. Finally, we show an additional advantage of SVM based learning technique – once SVM is trained, it can encode the information in the data effectively and answer several queries of interest.

1.2. Overview of the approach

An overview of our approach is presented in Fig. 1 for the design problem. With training data, a coarse grid search is performed to select the appropriate kernel. In the next step, a fine grid search is performed to select the parameters for the best kernel SVM. This makes the entire process generic for even a novice user to get best results by using SVM. The trained SVM can then be queried using PSFOD algorithm with test data. Parametric Search For Optimal Design (PSFOD) algorithm uses bisection method to find the optimal value of design parameters for best performance. A similar mechanism of training and testing is used for vibrating string problem.

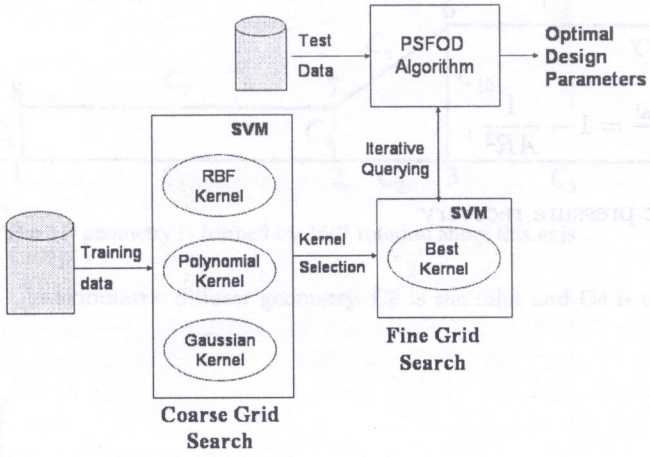

Fig. 1. An overview of SVM based framework

1.3. Organization of the paper

The rest of the paper is organized as follows. Section 2 discusses the problem domains and the need for the SVM based framework. The algorithm PSFOD is also discussed in this section. Section 3 describes SVM and our grid searching algorithm that automatically selects SVM parameters optimally. Experimental results and comparison with other approaches are presented in Section 4 followed by a discussion on advantages of SVM over Neural Networks. Conclusions and the scope for future work follow in Section 5.

2. Problem domain: diffuser design and oscillating string

2.1. Design Issues in a Diffuser

2.1.1. Introduction

A diverging passage in which subsonic flow decelerates between inlet and outlet is called a diffuser. The diffuser converts a portion of the kinetic energy of the fluid into static pressure. The problem geometry along with the diffuser parameters are shown in Fig. 2.

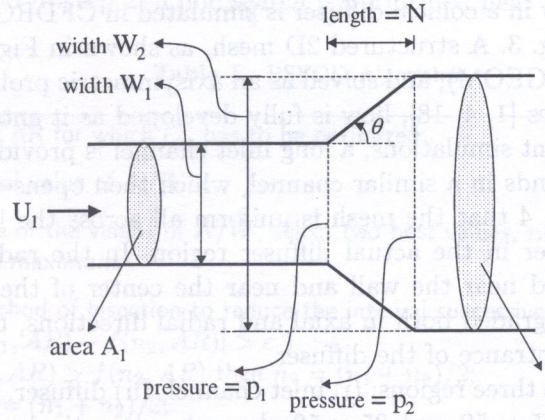

Fig. 2. View of a diffuser. Diffuser parameters are as follows: divergence angle (2θ), centerline length (N), inlet velocity (U_1), area ratio (A_2/A_1), and non-dimensional diffuser length (N/W_1)

The performance parameters associated with the diffuser design are as follows:

- ideal pressure recovery

$$C_{pi} = \frac{(p_2 - p_1)_{\text{ideal}}}{\rho U_1^2/2} = 1 - \frac{1}{AR^2}\,, \tag{1}$$

- overall (actual) static pressure recovery

$$C_p = \frac{p_2 - p_1}{\rho U_1^2/2}\,, \tag{2}$$

- diffuser effectiveness

$$\eta = \frac{C_p}{C_{pi}}\,. \tag{3}$$

The critical performance parameter controlling diffuser design is the pressure recovery, C_p. The choice of diffuser geometry parameters, viz. Area Ration (AR), Nondimensional diffuser length (N/W), diffuser size and its shape determine the pressure recovery. The standard methods for diffuser design involve varying these parameters so that optimum value of C_p is achieved. An ideal diffuser design will provide efficient pressure recovery over a wide range of flow conditions; however, this is rarely achieved in practice. Most designs are based on limited empirical data. The typical values of diffuser geometry parameters are: $\theta = 3°$ to $45°$, $N/W_1 = 1.5$ to 60 and $AR = 2$ to 100. Thus, it can be easily observed that the search space for performance optimization requires large number of computation.

The objective of this work is to employ SVM for predicting the value of C_p to sufficient degree of accuracy based on training with some sample values of design parameters. The training and prediction steps are very fast in SVM and thus the SVM-based design process is very efficient and accurate. In order to train the SVM, we generate a small set of values of C_p for varying range of N/W and AR. These values are generated using the CFDRC software. The trained SVM can be used to predict either of the following: (a) value of C_p for the intermediate values of N/W and AR, (b) value of AR for given N/W to obtain optimal C_p, (c) value of N/W for given AR to obtain optimal C_p.

2.1.2. Geometry of the diffuser

Numerical methods offer the potential to provide parametric studies for the designer to maximize the performance. Turbulent flow in a conical diffuser is simulated in CFDRC. The diffuser geometry is axisymmetric as shown in Fig. 3. A structured 2D mesh, as shown in Fig. 4, is generated for the diffuser in CFDRC (using CFD-GEOM), and solved as an axisymmetric problem (using CFD-ACE). In all previously reported studies [1, 4, 18], flow is fully developed as it enters the diffuser inlet. In order to ensure this in the present simulations, a long inlet channel is provided in the diffuser mesh. The outlet of the diffuser also ends in a similar channel, which then opens to the atmosphere.

It can be observed from Fig. 4 that the mesh is uniform all across the length (axial direction), although the resolution is higher in the actual diffuser region. In the radial direction, the mesh is graded, i.e., it is concentrated near the wall and near the center of the diffuser. In a previous published work [4], the mesh is graded both in axial and radial directions, being concentrated near the diffuser wall and near the entrance of the diffuser.

The mesh can be divided into three regions: (i) inlet channel, (ii) diffuser, and (iii) outlet channel. These regions have 100×50, 75×50 and 25×50 elements respectively in the axial and radial directions. This ensures that the inlet and outlet channel regions have the same concentration of elements, while diffuser region has a higher concentration.

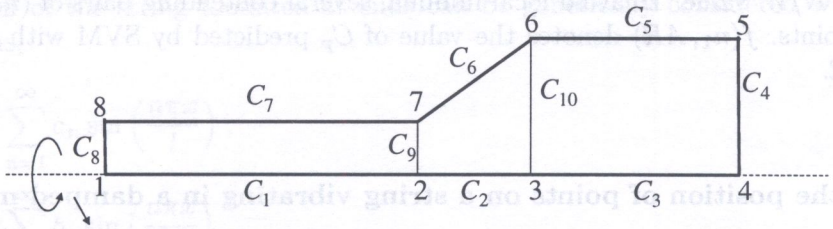

Fig. 3. Axisymmetric diffuser geometry. C8 is the inlet and C4 is the outlet

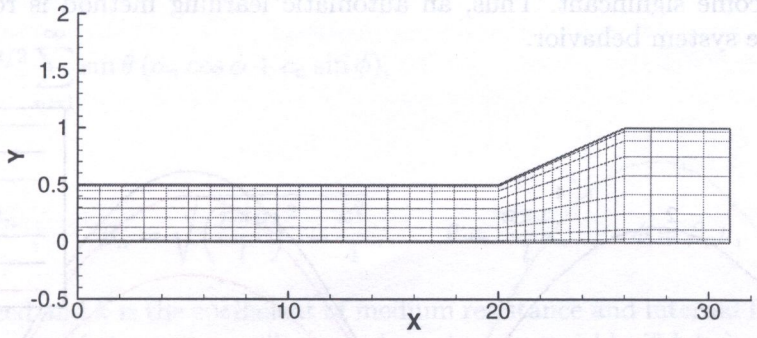

Fig. 4. Simplified view of the diffuser mesh

2.1.3. C_p maximization

To solve the maximization problem a new algorithm needs to be designed that can search the parametric space formed by non-dimensional diffuser length and area ratio. As the training points are discrete, the algorithm should cater to the problem of inherent discretization that results from training on discrete data points. Since SVM is expected to give less generalization error, the optimum values can be approximately selected especially if they lie between the discrete data points. The algorithm used is Parametric Search For Optimal Design (PSFOD) and is presented in Table 1. The constant ε in step 2 is assigned a small value. The algorithm is based on the hypothesis that the optimal value of N/W generally lies between the two values for which C_p is maximum. This hypothesis is supported by the fact that the underlying mathematical function over the parameters is smooth. In step 1, from the training database, two values of N/W (say n_1 and n_2) are selected for which C_p is maximum and in step 2, a fine search is performed near these two values, n_1 and n_2, to

Table 1. PSFOD Algorithm

Input: The value of AR for which C_p has to be optimized.

Output: The optimal value of N/W.

Step 1: In the range of the values of N/W, select two best values, n_1 and n_2 of N/W for which the value of C_p is maximum.

Step 2: Use the method of bisection to reduce the interval successively to half as shown below:

While ($|f(n_1, AR) - f(n_2, AR)| > \varepsilon$)

{ If $f(n_1, AR) > f(n_2, AR)$ then $n_2 = (n_1 + n_2)/2$;

else $n_1 = (n_1 + n_2)/2$;

}

Step 3: Report the value n_1 as the optimal

look for the best N/W value. To avoid local minima, several contending pairs of (n_1, n_2) are chosen as initial seed points. $f(n_1, AR)$ denotes the value of C_p predicted by SVM with $N/W = n_1$ and Area ratio $= AR$.

2.2. Locating the position of points on a string vibrating in a damped medium

Several solid mechanics systems oscillate under damped conditions and to mathematically model the dampening function and predict system response is a challenging problem [6, 8]. In general, such dampening models are not very accurate because of the inherent assumptions in the modeling process [7]. In real-life, the dampening function may dynamically change. In addition, new dampening factors may become significant. Thus, an automatic learning method is required that can dynamically model the system behavior.

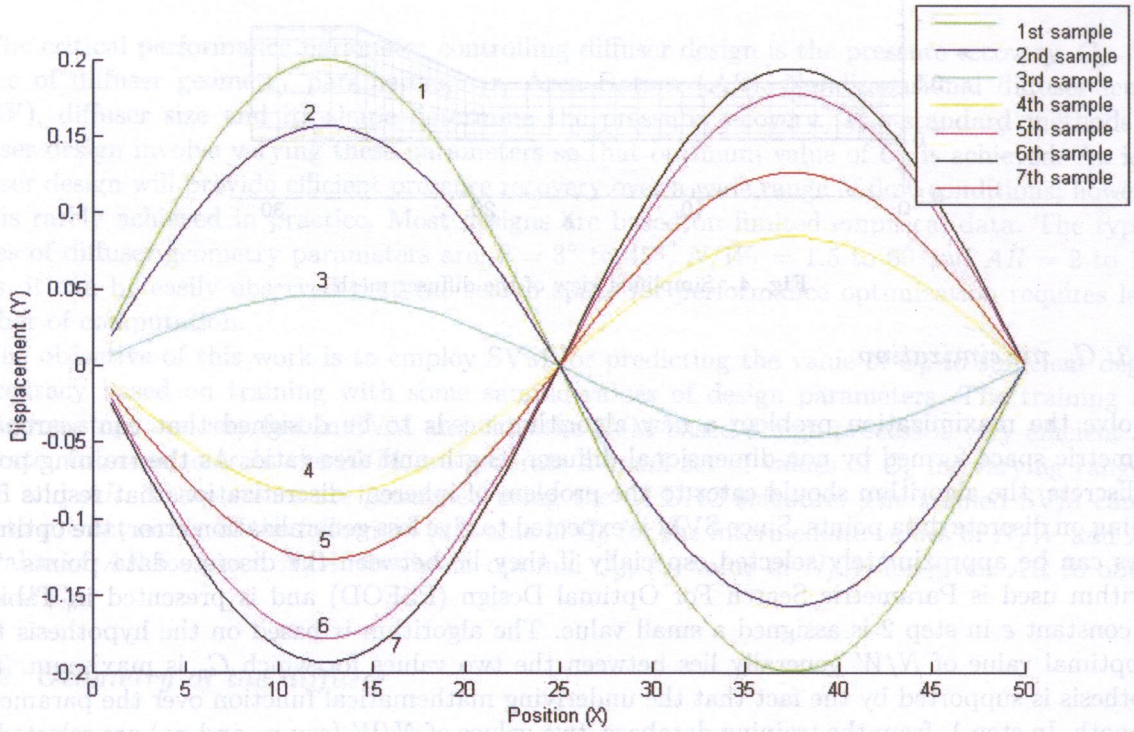

Fig. 5. Position of the string at different time samples

We propose an SVM based framework that continually updates its machine parameters incorporating recent changes in order to predict system response. In order to illustrate the working of our framework, a standard and well formulated problem of a string vibrating in damped medium is chosen (see Fig. 5). SVM is trained with input vectors consisting of string vibrations at different times taken from the observations. After training, SVM predicts the positions of vibration at other times. Thus, the string vibration problem can be formulated as:

Input to SVM: The position of different locations of a string vibrating with damping at various time instants.

Output of SVM: thee transverse position of the different locations of the string at any time.

The vibration of the string mounted at ends can be generically considered with the following initial conditions,

$$y(x,0) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi x}{l}\right), \tag{4}$$

$$\left(\frac{\partial y}{\partial t}\right)_{t=0} = \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{l}\right), \tag{5}$$

where y is the displacement, l is the length of the string, x is the position on the string, and a_n and b_n are the Fourier coefficients. The general equation guiding the vibration of string with appropriate initial conditions [2] is given below,

$$y(x,t) = e^{-kt/2} \sum_{n=1}^{\infty} \sin\theta \,(a_n \cos\phi + c_n \sin\phi), \tag{6}$$

where

$$c_n = \frac{b_n + \frac{k}{2}a_n}{\beta_n}, \qquad \beta_n = \sqrt{\left(\frac{n\pi c}{l}\right)^2 - \frac{k^2}{4}}, \qquad \theta = \frac{n\pi x}{l}, \qquad \phi = \beta_n t, \tag{7}$$

c is the wave speed and k is the coefficient of medium resistance and internal friction. The amplitude $y(x,t)$, the response of the string, will come closer to zero quickly if k is large. It can be observed that the complexity of the problem increases when the dampening function is not known *a priori*.

3. Support vector machines

3.1. Preliminaries

In MLT, two parameters denoted in terms of error are important in evaluating the learning performance. The first is the *training error* which is the frequency of errors made by a learning machine during the training session. The second is the *generalization error* which is defined as the frequency of errors made by the machine when it is tested with examples not seen before. The most frequently used measure for generalization error is root mean square error [11].

The support vector machine is a linear learning tool deriving its desirable property from the statistical learning theory. SVM is based on the idea of structural risk minimization, which bounds the generalization error to the sum of training set error and a term depending on the Vapnik-Chervonenkis dimension [25] of the learning machine. The SVM induction principle minimizes an upper bound on the error rate of a learning machine on test data (i.e., generalization error) rather than minimizing the training error itself which is used in empirical risk minimization. This helps them to generalize well on the unseen data.

Our approach involved transform data to the format of an SVM regression tool by conducting simple scaling on the data. Next we select the kernel and use coarse Grid-search to find the best parameters. We use the best kernel to train the whole training set and after testing, we perform fine grid-search for further accuracy.

3.2. Training SVM

Consider the problem of regression and function approximation where the learning machine is given N training data from which it tries to learn the input-output relationship (dependency, mapping or function) $f(x)$. Let the training data set be $D = \{[\mathbf{x}_i, y_i] \in \mathbf{R}^n * \mathbf{R},\ i = 1, \ldots, N\}$, where the

inputs $\mathbf{x}$ are n-dimensional vectors $\mathbf{x} \in \mathbf{R}^n$ and system responses $y \in \mathbf{R}$ are continuous values. The SVM considers approximating functions of the form

$$f(\mathrm{x},\mathrm{w}) = \sum_i w_i \mathbf{\Phi}_i(\mathbf{x})$$

where $\mathbf{\Phi}(\mathbf{w})$ is the cost function. Consider the ϵ-insensitivity loss function defined as follows,

$$\epsilon(p,\varepsilon) = \begin{cases} 0; & |p| \le \varepsilon, \\ |p| - \varepsilon; & |p| > \varepsilon. \end{cases} \tag{8}$$

Considering the ϵ-insensitivity loss function $\epsilon(y - f(x,w),\varepsilon)$, the loss is equal to 0 if the difference between the predicted $f(\mathbf{x},\mathbf{w})$ and the measured values is less than ε.

Vapnik's ϵ-insensitivity loss function defines an ϵ-tube around $f(\mathbf{x},\mathbf{w})$. If the predicted value is within the tube, the loss (i.e., error) is zero. For all other predicted points outside the tube, the loss equals the magnitude of the difference between the predicted value and the radius ϵ of the tube.

Now minimizing risk R equals

$$R_{\mathbf{w},\xi,\xi^*} = \left[\frac{\|w\|^2}{2} + C\left(\sum_i \xi_i + \sum_i \xi_i^*\right)\right] \tag{9}$$

for $i = 1, 2, \ldots, N$. C is the penalty parameter. The constraints can be formulated as follows,

$$\begin{aligned} &y_i - \mathbf{w}^T\mathbf{x}_i - b \le \epsilon + \xi_i, && \xi_i \ge 0, \\ &\mathbf{w}^T\mathbf{x}_i + b - y_i \le \epsilon + \xi_i^*, && \xi_i^* \ge 0, \end{aligned} \qquad \text{for } i = 1, 2, \ldots, N, \tag{10}$$

where ξ and ξ^* are slack variables for measurements 'above' and 'below' an ϵ-tube, respectively, as shown in Fig. 6 [13]. Both slack variables have positive values. Lagrange multipliers α_i and α_i^* corresponding to ξ and ξ^* will be nonzero values for training points 'above' and 'below' an ϵ-tube, respectively. Because no training data can be on both sides of the tube, either α_i or α_i^* will be non zero. For data points inside the tube, both multipliers will be equal to zero.

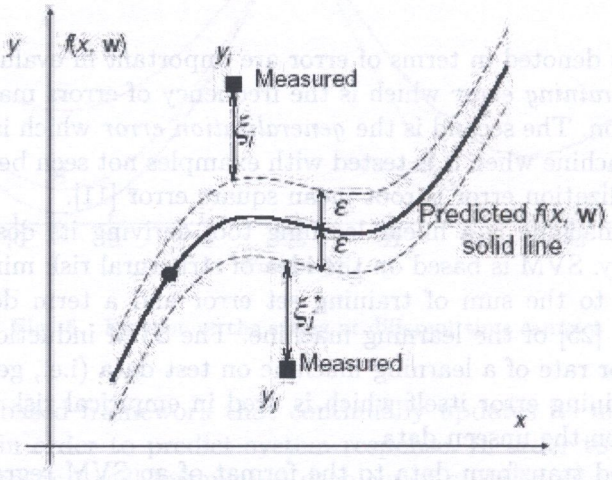

Fig. 6. The depiction of ϵ-insensitivity loss function

We solve this constrained optimization problem by constructing a *primal variables Lagrangian function* $J_p(\mathbf{w},\xi,\xi^*)$,

$$\begin{aligned} J_p(w,b,\xi,\xi^*,\alpha,\alpha^*,\beta,\beta^*) = 0.5\cdot\mathbf{w}^T\mathbf{w} + C\left(\sum_i \xi_i + \sum_i \xi_i^*\right) - \sum_i \alpha_i\left[\mathbf{w}^T\mathbf{x}_i + b - y_i + \epsilon + \xi_i\right] \\ + \sum_i \alpha_i^*\left[y_i - \mathbf{w}^T\mathbf{x}_i - b + \epsilon + \xi_i^*\right] - \sum_i \left[\beta_i\xi_i + \beta_i^*\xi_i^*\right]. \end{aligned} \tag{11}$$

The primal variables Lagrangian $J_p(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \beta, \beta^*)$ has to be *minimized* with respect to primal variables $\mathbf{w}$, b, ξ and ξ^* and *maximized* with respect to non-negative Lagrange multipliers α, α^*, β and β^*. This problem can be solved again either in a primal space or in a dual one.

Below, we consider a solution in a dual space as it has a faster convergence. Applying Karush–Kuhn–Tucker (KKT) conditions for regression, we maximize a dual variables Lagrangian $J_d(\alpha, \alpha^*)$ as follows,

$$J_d(\alpha, \alpha^*) = \sum_i y_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_i (\alpha_i - \alpha_i^*) - 0.5 * \sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_i^T \mathbf{x}_j \,, \tag{12}$$

subject to the following constraints,

$$\sum_i (\alpha_i - \alpha_i^*) = 0, \qquad 0 \le \alpha_i \le C, \qquad 0 \le \alpha_i^* \le C \qquad \text{for } i = 1, 2, \ldots, N. \tag{13}$$

Note that a dual variables Lagrangian $J_d(\alpha, \alpha^*)$ is expressed in terms of Lagrange multipliers α and α^* only. After training SVM, the test data point can be given as input and SVM predicts a value as output.

3.3. Parameter and kernel selection: The grid-search algorithm

The performance of SVM is strongly related to the choice of the kernel function and the penalty parameter C. There are a large number of kernel functions available such as linear kernel, polynomial kernel and Radial Basis Function (RBF) kernel. The non-linear kernels such as Gaussian, RBF, etc., non-linearly map samples into a higher dimensional space, and can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF as [14] shows that the linear kernel with a penalty parameter C has the same performance as the RBF kernel with some parameters (C, γ).

Since it is not known a-priori, which (C, γ) pairs would result in the best classification performance, an automated parameter selection technique would be highly preferred. Usually, these parameters are selected on a trial and error basis. For finding the optimum values of parameters (C, γ) automatically, a grid search technique is used using cross validation. In the k-fold cross-validation, the training set is first divided into k-sets of approximately equal size. The SVM is trained using the data from some subsets and its accuracy is determined using the left out set. We employ cross-validation to prevent the overfitting problem.

Our two-phase grid search algorithm proceeds as follows. In the first phase, a set of (C, γ) is used for evaluation of best kernel amongst Gaussian, RBF, etc. In the second phase, a larger number of pairs of (C, γ) are tried on best kernel and the one with the best cross-validation accuracy is picked. The grid search is performed in a hierarchical manner. Initially, the values of C and γ are grown exponentially. Keeping one of the parameters fixed, the other parameter is grown exponentially and the prediction performance is evaluated using cross validation. The two dimensional grid of interest is raster scanned and the region where the performance is the best is selected. After identifying a better region on the grid, a finer grid search on that region can be conducted. The grid search can easily be parallelized and although it is exhaustive, yet the computational time is not huge because of just two search parameters.

4. Experimental results and comparison

4.1. C_p function approximation

Two independent parameters of the conical diffuser geometry, Area ratio ($AR = A_2/A_1$) and Non-dimensional diffuser length (N/R_1), are varied. The parameter values for the geometries studied in this work are given in Table 2.

Table 2. The value of diffuser parameters used in experiments

Diffuser parameter	Values
Length of the inlet channel	20
Length of the outlet channel	5
Dimensionless diffuser length N/R_1	2, 2.5, 3, 3.5, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18, 20
Area Ratio A_2/A_1	1.05, 1.15, 1.25, 1.35, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3, 3.5, 4, 4.5, 5

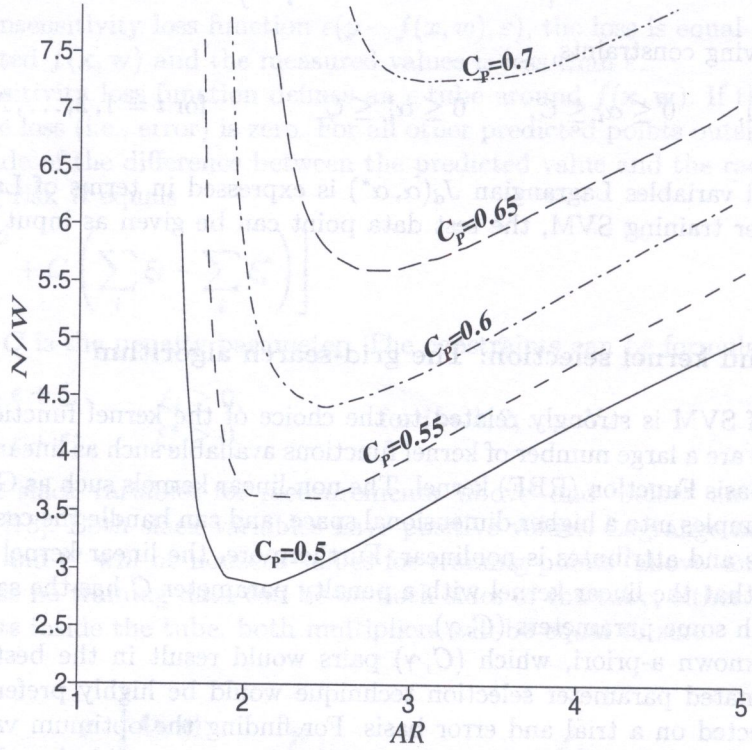

Fig. 7. The curves of N/W versus AR for constant C_p

Using the trained SVM, we obtain curves for C_p =const for given N/W and AR parameters as depicted in Fig. 7. The Reynolds number used is 5×10^4. The methodology employed is to give constant C_p and various values of AR as inputs to SVM and obtain values of N/W as output from SVM. These curves shown in Fig. 7 match qualitatively with the ones given in the literature [26] validating the SVM learning.

In order to do comparison between SVM and ANN, the data was generated for different Reynolds numbers as shown in Table 3. For ANN, three layer feedforward structure with 10 hidden neurons was taken. The transfer function used was tansig and the learning function used was Levenberg–Marquardt backpropagation [11]. The ANN learning parameters were decided based on several training runs and the parameters which gave the best performance were selected. The (C, γ) for RBF kernel SVM selected using grid search was (250, 0.3). The ε value used in the algorithm is 0.0001. In all cases, the training set consisted of 150 points with N/W and AR as inputs to SVM or ANN. The test set consisted of 75 cases for which C_p value taken as output from SVM or ANN was compared with known C_p values. The error measure used are Root Relative Mean Square Error (RRMSE) and Maximum Relative Error (MRE). Though Root Mean-Squared Error (RMSE) is the commonly used measure of accuracy of numeric prediction, it does not distinguish between whether the relative error is large for an actual value of a test case. The root relative mean square error (RRMSE) standardizes the RMSE computed per cell to the *true* value observed in that cell location. The resulting RRMSE value is expressed as a percent and represents the standard variation

of the estimator. The RRMSE assigns equal weight to any overestimation or underestimation of the statistic. RRMSE is expressed as [15],

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{N}\left[\frac{Y_i'-Y_i}{Y_i}\right]^2}{N}}, \tag{14}$$

where, Y_i is the actual value of the i-th sample, Y_i' is the predicted value and N is the total number of samples. Similarly, MRE can be calculated. It can be observed from Table 3 that SVM shows better performance on both RRMSE and MRE measures. The SVM performance is at least two times better than ANN in all cases.

Table 3. Comparison of SVM with ANN for predicting C_p values in diffuser design problem

Reynolds Number	Root Relative Mean Square Error		Maximum Relative Error	
	SVM	ANN	SVM	ANN
5×10^4	0.0016	0.0047	0.0522	0.2881
1×10^5	0.0371	0.0684	2.7757	5.1139
3×10^5	0.0028	0.0193	0.1011	1.3495
5×10^5	0.0027	0.0045	0.1336	0.1926

4.2. Design using PSFOD algorithm

SVM can be applied to solve the design problems such as finding the optimum value of N/W and AR to maximize pressure recovery. Figure 8 shows the plot for optimal value of AR given N/W to maximize pressure recovery. The optimal values are obtained by employing PSFOD algorithm. This

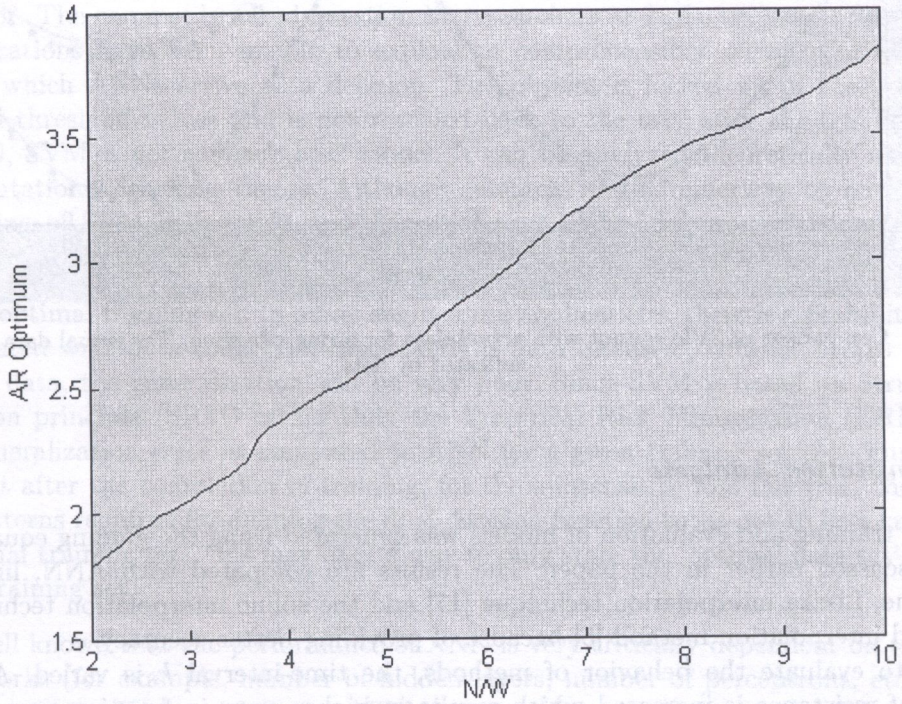

Fig. 8. Plot showing the optimum value of AR for given value of N/W

figure can serve as a reference chart for an engineer. For instance, consider for an application the desired N/W is 6. Using Fig. 8, it is obvious that the designer needs to concentrate on AR values close to 2.9 for maximizing pressure recovery.

4.3. String problem

4.3.1. Qualitative analysis

Figure 9 shows the actual location of points on the string as compared with the ones predicted by the SVM. It can be observed that the prediction done by SVM matches closely with the original displacements. We can also infer from this graph that both RRMSE and MRE play their part in estimating the accuracy of the model. While RRMSE gives an idea of how close the overall method was to the actual results, MRE gives an idea of how accurately the position of a particular point can be determined.

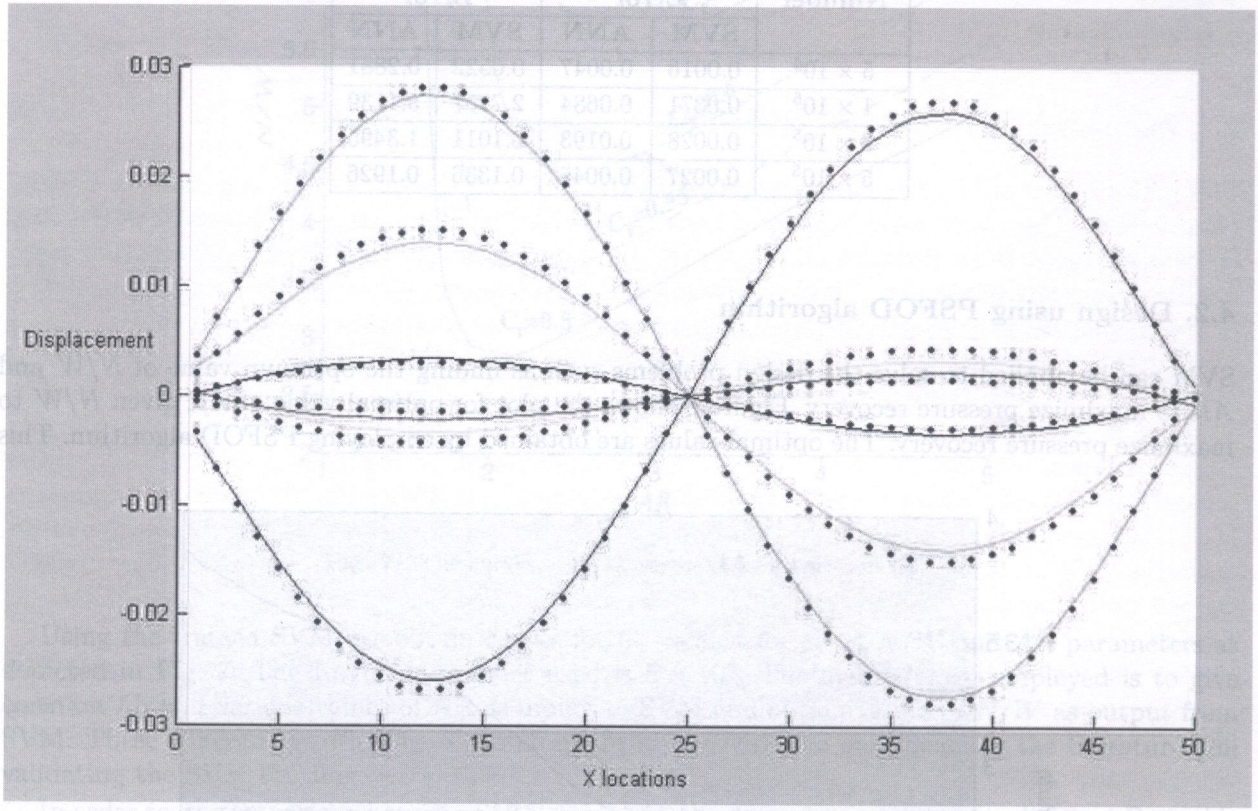

Fig. 9. Comparison of SVM output with actual data for string vibration. The actual data points are indicated by dots

4.3.2. Quantitative Analysis

The data for training and evaluation of models was generated using the guiding equations of string vibration discussed earlier in the paper. The results are compared with ANN, linear interpolation technique, Liszka interpolation technique [17] and the spline interpolation technique, which is a widely used interpolation method [5] because of providing better accuracy.

In order to evaluate the behavior of methods, the time-interval k is varied. As k increases, the impact of resistance is increased which results in higher error in function approximation. We consider the cases with varying k and varying time steps to judge the generalization performance

Table 4. Comparison of SVM with other methods for varying time step and parameter k. MRE stands for Maximum Relative Error and RRMSE for Root Relative Mean Square Error

	Error	SVM	Spline Interpolation	Linear Interpolation	ANN	Liszka Interpolation
Case1	RRMSE	0.0011	0.0085	0.0475	0.0730	0.1366
($k = 0.2$, $t = 1/200$)	MRE	0.0757	0.3219	1.4309	0.3703	2.6919
Case2	RRMSE	0.0210	0.6077	0.3290	0.1405	0.4203
($k = 0.2$, $t = 1/100$)	MRE	0.5808	1.778	2.8869	2.7528	2.2847
Case 3	RRMSE	0.0085	0.0602	0.3863	0.1068	0.0938
($k = 0.5$, $t = 1/300$)	MRE	0.1012	0.5570	3.1462	2.7248	2.9681
Case 4	RRMSE	0.0213	0.1292	0.1235	0.1468	0.1836
($k = 0.5$, $t = 1/200$)	MRE	0.3116	1.031	1.076	0.7142	1.4224
Case 5	RRMSE	0.377	0.6406	1.021	0.6711	0.3901
($k = 1$, $t = 1/100$)	MRE	3.243	6.4889	3.458	5.827	5.0326

of the learning methods as shown in Table 4. For the case, $k = 0.5$ with time steps size being $1/200$, the samples were obtained at 0, 0.005, 0.01, 0.015, 0.020 and so on. The other parameters were assigned the following values: $l = 5$; $c = 80$; $a = 0.05$; frequency = 16 Hz. Table 4 shows the comparison of SVM with other methods. It can be easily observed that the SVM results are much better than those obtained using the interpolation techniques. It can be seen that with increasing k and t, the RRMSE and MRE of methods other than SVM show significant increase. Consider case 4 ($k = 0.5$ and $t = 1/200$), the RRMSE of SVM is 0.0213, while RRMSE of other methods is above 0.12, which is a significant and in many cases, an intolerable error.

4.4. Comparison with Neural Networks

There are several advantages of SVM over ANN as listed below:

1. ANN does not reveal any physics and after learning it is difficult to interpret the learning weights and behavior. The commonly asked question by researchers and practitioners alike is that most ANN applications have been unable to explain in comprehensibly meaningful way the basic process by which ANNs arrive at a decision. The physics is locked up in the set of optimal weights and threshold values and is not revealed back to the user after the training. However, unlike ANN, SVM is not a 'black box' model. It can be analyzed theoretically using concepts from computational learning theory. Although research is still underway to have a complete understanding of SVM in function approximation, yet SVM offers some basic explanation on how it arrives at the final decision.
2. Identifying optimal training set: In many engineering applications, there is a prohibitive cost and time associated with data collection. Since ANN is data intensive, without proper quality and quantity of data, the generalization will be very poor. Since SVM is based on Structural Risk Minimization principle (SRM) rather than the Empirical Risk Minimization (ERM), it offers a better generalization error as compared to ANN for a given training set [24]. Further, it can be seen that after the completion of training, for the e-insensitive loss function, the number of training patterns required for defining the final decision function turns out to be a small fraction of the original training set. This may offer a way to only store the 'optimal data set' rather than the whole training set.
3. It is also well known that the performance of ANN is very critically dependent on the structure of the networks (for example, number of hidden layers, number of perceptrons, etc.) in Multi-layered perceptrons. However, in case of SVM, recent researches [21] suggest that the structure of SVM can be closely pre-decided using the training data analysis.

5. Conclusions

In the field of mechanics, most of the experiments take a large amount of time and require huge setups. The theoretical models for unknown parameters like roughness, dampening, etc., also do not always match perfectly with the observed results. Our work encompasses the development and application of unbiased learning technique used by SVM to the studies in this field. There are many possible applications specifically in the field of fluid and solid mechanics. We have explored two of them: conical diffuser design problem and prediction of position of a vibrating string.

The goal of SVM modeling is to choose a model from the hypothesis space, which is closest (with respect to some error measure) to the underlying function in the target space. SVM has been shown to yield better results than the conventional methods like interpolation and it is able to work with training sets of relatively smaller size than the ones used by other methods like ANN. The prediction of values at data points using SVM is much faster than the generation of data points using traditional modeling softwares like CFDRC. So it is desirable to use SVM for both, their accuracy and speed.

The methodology presented in the paper can be easily extended to other design problems and function approximation problem in engineering applications. The grid-searching method for selecting best SVM model and PSFOD algorithm used in optimal design process can prove especially helpful in solving such problems efficiently.

Future work

In our work, the optimum SVM parameters were selected using grid searching. For applications where sufficient data can be generated, a better approach would be to use analytic parameter selection directly from the training data [3], rather than re-sampling approaches commonly used in most SVM applications. In this paper, optimal design and parameter prediction problems were tackled. Recent advances in SVM [21] has shown the use of Least Squares SVM (LS-SVM) for the optimal control of nonlinear systems. The use of LS-SVM can be explored to mechanics problems where feedback and control is required in real-time.

References

[1] R.S. Azad. Turbulent flow in a conical diffuser: a review. *Experimental Thermal and Fluid Science*, **13**: 318–337, 1996.

[2] D.R. Bland. *Wave Theory and Applications*, ch. 2, pp. 61–70. Clarendon Press, Oxford, 1988.

[3] V. Cherkassky, Y. Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Int. Journal of Neural Networks*, 0893-6080, **17**(1): 113–126, 2004.

[4] N.M. Cho, C.A.J. Fletcher. Computation of turbulent conical diffuser flows using a non-orthogonal grid system. *Computers and Fluids*, **19**: 347–361, 1991.

[5] A. Davies, P. Samuels. *An Introduction to Computational Geometry for Curves and Surfaces.* Clarendon Press, 1996.

[6] B.L. Fowler, E.M. Flint, S.E. Olson. Design methodology for particle damping. *The SPIE Conference on Smart Structures and Materials*, Newport Beach, CA, March 4–8, pp. 4331-20, 2001.

[7] R.W. Fox, A.T. Mcdonald. *Introduction to Fluid Mechanics*, Ed. 4, ch. 8, pp. 352–354. John Wiley and Sons, 1995.

[8] R.M. Glaese, E.H. Anderson. Vibration control using passive and active resonant devices. *SPIE Smart Structures Conference*, 2005.

[9] P. Hajela, L. Berke. Neurobiological computational models in structural analysis and design. AIAA-90-1133-CP, *31st SDM Conf.*, Baltimore, MD, April 8–10, pp. 335–343, 1991.

[10] F.M. Ham, I. Kostanic. *Principles of Neurocomputing for Science and Engineering.* McGraw–Hill, 2001.

[11] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 1998.

[12] D.-S. Jeng, D.H. Cha, M. Blumenstein. Application of Neural Network in Civil Engineering Problems, *The International Internet-Processing-Systems-Interdisciplinaries (IPSI-2003) Conference*, Seveti-Stefan, Montenegro, October 5–11, 2003.

[13] V. Kecman. *Learning and soft computing Support Vector Machines, neural networks and fuzzy logic models.* The MIT press, Cambridge, MA, 2001.

[14] S.S. Keerthi, C.J. Lin. Asymptotic behaviors of Support Vector Machines with Gaussian kernel. *Neural Computation*, 1667–1689, 2003.

[15] C. Kroll, J. Stedinger. Estimation of moments and quantiles using censored data. *Water Resources Research*, **32**(4): 1005–1012, 1996.

[16] H. Lee, L. Hajela. Prediction of turbine performance using MFN by reducing mapping nonlinearity. In: G.H.V. Topping, M. Papadrakakis, eds., *Artificial Intelligence and Object Oriented Approaches for Structural Engineering*, pp. 99–105, 1995.

[17] T. Liszka. An interpolation method for an irregular net of nodes. *Int. J. Num. Meth. Engrg.*, **20**: 1599–1612, 1984.

[18] A.T. McDonald, R.W. Fox. An experimental investigation of incompressible flow in conical diffusers. *Int. J. Mech. Sci.*, **8**: 125–139, 1966.

[19] S. Mukherjee, E. Osuna, F. Girosi. Nonlinear prediction of chaotic time series using a support vector machine. In: J. Principle, L. Gile, N. Morgan, E. Wilson, eds., *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, New York, 1997.

[20] S.J. Schreck, W.E. Faller, M.W. Luttges. Neural network prediction of three-dimensional unsteady separated flow fields. *J. Aircraft*, **32**: 178–185, 1995.

[21] J.A. Suykens, J. Vandewalle, B.D. Moor. Optimal control by least squares support vector machines. *Int. J. Neural Networks*, **14**(1): 23–35, 2001.

[22] A. Tzes, J. Borowiec. Applications of fuzzy logic and neural networks to identification and control problems in fluid mechanics. *ASME International Mech. Engineering Congress and Exposition*, Atlanta, Georgia, 1996.

[23] R.D. Vanluchene, R. Sun. Neural networks in structural engineering. *Microcomputers in Civil Engineering*, **5**(3): 207–215, 1990.

[24] V. Vapnik, S. Golowich, A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In: M. Mozer, M. Jordan, T. Petsche, eds., *Advances in Neural Information Processing Systems 9*, pp. 281–287. MIT Press, Cambridge, MA, 1997.

[25] V. Vapnik. *Statistical Learning Theory.* Springer, NY, 1998.

[26] W. Wysocki, Z. Kazimierski. Analysis of subsonic transitory stalled flows in straight-walled diffusers. *ASME J. Fluids Engineering*, **108**: 222–226, 1986.