MODELING AND MULTI-OBJECTIVE OPTIMIZATION OF STALL CONTROL ON NACA0015 AIRFOIL WITH A SYNTHETIC JET USING GMDH TYPE NEURAL NETWORKS AND GENETIC ALGORITHMS

R. Razaghi, N. Amanifard* and N. Narimanzadeh

Department of Mechanical Engineering, Faculty of Engineering, University of Guilan
P.O. Box 3756, Rasht, Iran
ramin_razaghi@yahoo.com - namanif@guilan.ac.ir – nnzadeh@guilan.ac.ir

*Corresponding Author

(Received: March 12, 2008 – Accepted in Revised Form: May 9, 2008)

Abstract This study concerns numerical simulation, modeling and optimization of aerodynamic stall control using a synthetic jet actuator. The numerical simulation was carried out by a large-eddy simulation that employs a RNG-based model as the subgrid-scale model. The flow around a NACA0015 airfoil, including a synthetic jet located at 10 % of the chord, is studied under Reynolds number Re = 12.7×10^6 and the angle-of-attack at 18-deg conditions. Then, group method of data handling (GMDH) type neural networks are used for modeling the effects of the actuators parameters (momentum coefficient, reduced frequency, angle with respect to the wall) on both developed time-averaged lift (C_L) and time-averaged drag (C_D), using some numerically obtained training and test data. To use the obtained polynomial neural network models, multi-objective genetic algorithms (GAs) (non-dominated sorting genetic algorithm, NSGA-II) with a new diversity preserving the mechanism, which is then used for Pareto based optimization of control parameters considers two conflicting objectives such as lift (C_L) and drag (C_D). It is shown that some interesting and important relationships as useful optimal design principles are involved in the performance of stall control on NACA0015 airfoil. Using a synthetic jet actuator can be discovered by the Pareto based multi-objective optimization of polynomial models. Such important optimal principles would not have been obtained without the use of both GMDH-type neural network modeling and Pareto optimization approach.

Keywords Aerodynamic Stall Control, Multi-Objective Optimization, Gas, Synthetic Jet

چکیده شبکه‌های عصبی از نوع GMDH و الگوریتم ژنتیک برای مدل‌سازی و بهینه‌سازی کنترل جدایی جریان ناپایدار حول ایرفویل با تزریق نوسانی مورد استفاده قرار گرفته‌اند. در ابتدا کنترل جدایی جریان برای ایرفویل NACA0015 در $Re = 12.7 \times 10^6$ و $Lj/C = 0.1$ با استفاده از مدل LES مورد بررسی عددی قرار گرفته و نتایج حاصله با نتایج تجربی مقایسه گردیده است. سپس با استفاده از داده‌های حاصل از شبیه‌سازی عددی، دو مدل ریاضی به کمک شبکه‌های عصبی از نوع GMDH برای مدل‌سازی ضریب برا و پسا با توجه به پارامترهای کنترل جدایی جریان (متغیرهای طراحی) مانند زاویه، شدت و فرکانس تزریق بدست آمده است. در انتها با استفاده از الگوریتم ژنتیک چند تابع هدفی مدل‌های بدست آمده از رفتار آیرودینامیکی ایرفویل مورد بهینه‌سازی قرار گرفته است. در بهینه‌سازی با دو تابع هدف (ضریب برا وپسا) از روش NSGAII استفاده شده و منحنی Pareto Front آن که نشان دهنده بهترین نقاط طراحی برای توابع هدف مورد نظر می‌باشد، رسم گردیده است. همچنین به کمک بهینه‌سازی با الگوریتم ژنتیک چند تابع هدفی (NSGAII)، برخی روابط مهم که می‌توان به عنوان قواعد طراحی بهینه موثر در عملکرد کنترل جدایی جریان مورد استفاده قرار داد، بدست آمده که بدون استفاده از روش مدل‌سازی با شبکه‌های عصبی GMDH و بهینه‌سازی با الگوریتم ژنتیک چند تابع هدفی NSGAII قابل حصول نبود.

1. INTRODUCTION

The manipulation of flow field around the airfoil to change the aerodynamic characteristics and improve performance of an airfoil and also prevent a flow from taking an undesired path is one of leading

technologies in the flow control areas. Indeed, the characteristics of an airfoil, such as lift, drag or pitching moment, may be adjusted using flow control strategies, without angle of attack modification or flap deflection. Therefore, active flow control may produce interesting approaches for a large variety of problems, e.g. changing lift for rotary wing aircraft [1], designing minimum radar cross-section aircraft or delaying aerodynamic stall to enhance maximum lift [2]. However, most of the active flow control techniques based on steady jet suction or blowing suffer from technical complexity and additional parts, thus the weight as well as additional maintenance result from plumbing systems.

Recently, an innovative actuator, called synthetic jet and based on high frequency zero-net-mass-flux injection, was tested experimentally [3-5]. Synthetic jet actuators are self-conditioned, small, light and energy-efficient. Moreover, they offer significant benefits over oscillatory blowing techniques since they do not depend on a supply of injected flow and require only electrical power input and so may be easily implemented in practical airfoils. Its capability to increase the lift for a cylinder in cross flow was demonstrated [6], whereas post-stall lift increase for symmetric airfoils was also reported [3]. The control parameters, such as the frequency actuation, the location of the blowing slot and the momentum coefficient were also investigated [2,3].

Numerical prediction of the beneficial effects of flow control reported in experimental studies was the subject of the investigation by Wu, et al [7,8] where flow control was simulated by a pulsating jet, which was located at quarter chord. It was found that lift increase in the post-stall regime could be achieved as was reported in the experiments. The same approach was employed by Hassan, et al [1] for a slot at 13 % of the chord, for different amplitudes and frequencies. It was reported that with the careful selection of peak amplitude and oscillation frequency, the lift can be increased. Donovan [9] performed simulations of steady and pulsating jet flow controls and compared to experimental measurements performed by Seifert, et al [2]. The same configuration was also studied by Ekaterinaris [10], who tested some different jet parameters.

All these numerical studies intend to further increase the knowledge on flow control process and for understanding the connection between the actuator parameters and the control efficiency. However, there are few meaningful study available aimed at optimizing the parameters of synthetic jet, such as jet momentum coefficient, injection frequency and injection flow angle due to the complexity of such flows and the computational cost of simulating many cases. Moreover, optimization of control parameters is indeed a multi-objective optimization problem rather than a single objective optimization, which is considered to date. Both lift and drag in airfoil is the important objective functions to be optimized in such a real-world complex multi-objective optimization problem. As a result, for every new airfoil, the optimal value flow actuation parameters are determined heuristically and depend on trail and error and the designers' knowledge. Such approach is necessarily limited and time consuming, since each time a time-accurate computation has to be performed for each attempt. Therefore, modeling and optimization of the parameters of a synthetic jet is investigated in the present study, by using GMDH-type neural networks and multi-objective genetic algorithms in order to maximize the lift and minimize the drag.

System identification techniques are applied in many fields in order to model and predict the behaviors of unknown and/or very complex systems based on given input-output data [11]. In this way, soft-computing methods [12], which concern computation in an imprecise environment, have gained significant attention. The main components of soft computing are, fuzzy logic, neural network, and evolutionary algorithms have shown great ability in solving complex non-linear system identification and control problems. Many research efforts have been expended to use evolutionary methods as effective tools for system identification [13-18]. Among these methodologies, Group Method of Data Handling (GMDH) algorithm is a self-organizing approach by which gradually more complicated models are generated based on the evaluation of their performances on a set of multi-input, single-output data pairs (X_i,y_i) $(i = 1, 2,\ldots,M)$. The GMDH was first developed by Ivakhnenko, et al [19] as a multivariate analysis method for complex systems modeling and identification. In this way, the GMDH was used to circumvent the difficulty of having a priori knowledge of the mathematical model of the process being

considered. Therefore, GMDH can be used to model complex systems without having specific knowledge of the systems. The main idea of GMDH is to build an analytical function in a feed forward network based on a quadratic node transfer function [20] whose coefficients are obtained using the regression technique. In fact, the real GMDH algorithm in which the model coefficients are estimated by means of, least square method has been classified as complete induction and incomplete induction, which represent the combinatorial (COMBI) and multilayered iterative algorithms (MIA), respectively [21]. In recent years, however, the use of such self-organizing networks has lead to successful application of the GMDH-type algorithm in a broad range of areas in engineering, science and economics [19-25].

There have been many efforts in recent years to deploy population based stochastic search algorithms such as evolutionary methods to design artificial neural networks since such evolutionary algorithms are particularly useful for dealing with complex problems having large search spaces with many local optima [22,21,26]. A very comprehensive review of using evolutionary algorithms in the design of artificial neural networks can be found in reference [27]. Recently, genetic algorithms have been used in a feed forward GMDH type neural network for each neuron searching its optimal set of connection with the preceding layer [23,28]. In the former reference, the authors have proposed a hybrid genetic algorithm for a simplified GMDH type neural network in which the connection of neurons are restricted to adjacent layers. Such shortcoming has been recently removed by the work of some authors [29,30], which will be represented in the following sections.

Basically, the optimization process is defined as finding a set of values for a vector of design variables so that it leads to an optimum value of an objective or cost function. There are many calculus based methods including gradient approaches to single objective optimization and are well documented in references [31-32]. However, some basic difficulties in the gradient methods, such as their strong dependence on the initial guess, cause them to find local optima rather than global ones. The nature inspired Genetic Algorithms (GAs) [33-34] differ from other traditional calculus based techniques. The main difference is that GAs work with a population of candidate solutions, not a single point in search space. This helps significantly to avoid being trapped in local optima [35] as long as the diversity of the population is well preserved. Such an advantage of evolutionary algorithms is very fruitful to solve many real world optimal design or decision making problems, which are indeed multi-objective. In these problems, there are several objective or cost functions (a vector of objectives) to be optimized (minimized or maximized) simultaneously. These objectives often conflict with each other so that improving one of them will deteriorate another. Therefore, there is no single optimal solution as the best with respect to all the objective functions. Instead, there is a set of optimal solutions, known as Pareto optimal solutions or Pareto front [36-40] for multi-objective optimization problems. The concept of Pareto front or set of optimal solutions in the space of objective functions in multi-objective optimization problems (MOPs) stands for a set of solutions that are non-dominated to each other but are superior to other solutions in the search space. This means that it is not possible to find a single solution to be superior to all other solutions with respect to all objectives so that changing the vector of design variables in such a Pareto front consisting of these non-dominated solutions could not lead to improvement of all objectives simultaneously. Consequently, such a change will lead to deteriorating of at least one objective. Thus, each solution of the Pareto set includes at least one objective inferior to that of another solution in that Pareto set, although both are superior to others in the rest of the search space. The early use of evolutionary search is first reported in 1960s by Rosenberg [41]. Since then, there has been a growing interest in devising different evolutionary algorithms for MOPs. Among these methods, the Vector Evaluated Genetic Algorithm (VEGA) proposed by Schaffer [42], Fonseca and Fleming's Genetic Algorithm (FFGA) [37], the non-dominated Sorting Genetic Algorithm (NSGA) by Srinivas, et al [36], the Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler, et al [43], the Pareto archived evolution strategy (PAES) by Knowles, et al [44], and the Niched-Pareto Genetic Algorithm (NPGA) by Horn, et al [45] are the most important ones. A

very good and comprehensive survey of these methods has been presented in references [46-49].

Basically, both NSGA and MOGA as Pareto based approaches, use the non-dominated sorting procedure originally proposed by Goldberg [33]. There are two important issues that have to be considered in such evolutionary multi-objective optimization methods: driving the search towards the true Pareto front and preventing premature convergence or maintaining the genetic diversity within the population [49]. The lack of elitism was a motivation for the modification of that algorithm to NSGA-II [50] in which a direct elitist mechanism has been introduced to enhance the population diversity. This modified algorithm has been regarded as the state of the art in evolutionary MOPs [51]. A comparison study among SPEA and other evolutionary algorithms on several problems and test functions showed that SPEA clearly outperforms the other multi-objective EAs [52]. Some further investigations developed in reference [49] demonstrated, however, that the elitist variant of NSGA (NSGA-II) performed similar to SPEA. In addition to its popularity and effectiveness, NSGA-II has been modified by Nariman-zadeh, et al [29] to enhance its diversity preserving mechanism, which will be used in this work.

In this paper, the numerical methods used for the simulation of the flow are described in the first sections and applied to study the flow separation control over a NACA 0015 airfoil including a synthetic jet. The validation of the code is achieved by comparison of the results obtained in this research versus experimental data. Nearly 144 numerical simulations are performed over a range of parameters (intensity, frequency and angle). Next, genetically optimization GMDH-type neural networks are used to determine the effects of jet momentum coefficient and reduced frequency of injection on both lift and drag in different jet angel. The total numbers of experimental data are 144 from which 116 are used for training whilst the remaining 28 data are merely used for model evaluation. The obtained polynomial models are then used in a Pareto-based optimization approach to find the best possible combination of lift (C_l) and drag (C_d) known as the Pareto front. The corresponding variations of design variables, namely, momentum coefficient, reduced frequency and angle of injection known as the Pareto set constitute some important design principles which can be effectively applied to determine optimal control parameters of the synthetic jet.

2. MODELING USING GMDH TYPE NEURAL NETWORKS

By means of GMDH algorithm, a model can be represented as set of neurons in which different pairs in each layer are connected through a quadratic polynomial and, thus, produce new neurons in the next layer. Such representation can be used in modeling, to map inputs to outputs. The formal definition of the identification problem is to find a function $\hat{f}$ that can be approximately used instead of actual one, f in order to predict output $\hat{y}$ for a given input vector $X = (x_1,x_2,x_3,\ldots,x_n)$ as close as possible to its actual output y. Therefore, given M observation of multi-input, single output data pairs so that

$$y_i = f(x_{i1},x_{i2},x_{i3},\ldots,x_{in}) \quad (i = 1, 2\ldots,M), \tag{1}$$

It is now possible to train a GMDH type neural network to predict the output values $\hat{y}_i$ for any given input vector $X = (x_{i1},x_{i2},x_{i3},\ldots,x_{in})$, that is

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, x_{i3},..., x_{in}) \quad (i = 1, 2, \ldots, M), \tag{2}$$

The problem is now to determine a GMDH type neural network so that the square of difference between the actual output and the predicted one is minimized, that is

$$\sum_{i=1}^{M} [\hat{f}(x_{i1}, x_{i2}, x_{i3},..., x_{in}) - y_i]^2 \rightarrow \min \tag{3}$$

General connection between inputs and output variables can be expressed by a complicated discrete form of the Volterra functional series in the form of

$$y=a_0+\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \ldots \tag{4}$$

Where is known as the Kolmogorov-Gabor

polynomial [17-19]. This full form of mathematical description can be represented by a system of partial quadratic polynomials consisting of only two variables (neurons) in the form of

$$\hat{y} = G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \quad (5)$$

In this way, such partial quadratic description is recursively used in a network of connected neurons to build the general mathematical relation of inputs and output variables given in Equation 4. The coefficients a_i in Equation 5 are calculated using regression techniques [17,19] so that the difference between actual output, y, and the calculated one, $\hat{y}$ for each pair of x_i, x_j as input variables is minimized. Indeed, it can be seen that a tree of polynomials is constructed using the quadratic form given in Equation 5 whose coefficients are obtained in a least squares sense. In this way, the coefficients of each quadratic function G_i are obtained to fit optimally the output in the whole set of input-output data pair, that is

$$E = \frac{\sum_{i=1}^{M} (y_i - G_i())^2}{M} \rightarrow \min \quad (6)$$

In the basic form of the GMDH algorithm, all the possibilities of two independent variables out of total n input variables are taken in order to construct the regression polynomial in the form of Equation 5 that best fits the dependent observations (y_i, i = 1, 2, …, M) in a least squares sense. Consequently, $\binom{n}{2} = \frac{n(n-1)}{2}$ neurons will be built up in the first hidden layer of the feed forward network from the observations $\{(y_i, x_{ip}, x_{iq}); (i = 1, 2, \ldots, M)\}$ for different p, q∈ {1, 2, …., n}. In other words, it is now possible to construct M data triples $\{(y_i, x_{ip}, x_{iq}); (i = 1, 2, \ldots, M)\}$ from observation using such p, q∈ {1, 2, …., n} in the form:

$$\left[\begin{array}{cc|c} x_{1p} & x_{1q} & y_1 \\ x_{2p} & x_{2q} & y_2 \\ \hline x_{Mp} & x_{Mq} & y_M \end{array}\right]$$

Using the quadratic sub-expression in the form of Equation 5 for each row of M data triples, the following matrix equation can be readily obtained as

$$A\,a = Y, \quad (7)$$

Where a is the vector of unknown coefficients of the quadratic polynomial in Equation 5

$$a = \{a_0, a_1, a_2, a_3, a_4, a_5\}, \quad (8)$$

and

$$Y = \{y_1, y_2, y_3, \ldots, y_M\}^T, \quad (9)$$

Is the vector of output's value from observation. It can be readily seen that

$$A = \left[\begin{array}{cccccc} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ \hline 1 & x_{Mp} & x_{Mq} & x_{Mp}x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \end{array}\right] \quad (10)$$

The least-squares technique from multiple regression analysis leads to the solution of the normal equations in the form of

$$a = (A^T A)^{-1} A^T Y, \quad (11)$$

Which determines the vector of the best coefficients of the quadratic Equation 5 for the whole set of M data triples. It should be noted that this procedure is repeated for each neuron of the next hidden layer according to the connectivity topology of the network. However, such a solution directly from normal equations is rather susceptible to round off errors and, more importantly, to the singularity of these equations.

2.1. Application of SVD to the Design of GMDH- Type Neural Networks Singular Value Decomposition (SVD) is the method for solving most linear least square problems in which some singularities may exist in the normal equations. The SVD of a matrix, $A \in \Re^{M \times 6}$ is a factorization of the matrix into the product of three matrices, column-orthogonal matrix $U \in \Re^{M \times 6}$, diagonal matrix $W \in \Re^{6 \times 6}$ with non-negative elements (singular values), and orthogonal matrix

$V \in \Re^{6\times 6}$ such that:

$$A = U\ W\ V^T \quad (12)$$

The problem of optimal selection of vector of the coefficients in Equations 8 and 11 is firstly reduced to finding the modified inversion of diagonal matrix W (in which the reciprocals of zero or near zero singulars (according to a threshold) are set to zero). Then, such optimal a is calculated using the following relation:

$$a = V\ [diag(1/w_j)]\ U^T\ Y \quad (13)$$

Such parametric identification problem is part of the general problem of modeling when structure identification is considered together with the parametric identification problem simultaneously. In this work, a new encoding scheme is presented in an evolutionary approach for simultaneous determination of structure and parametric identification of GMDH neural networks.

2.2. Application of GA in the Topology Design of GMDH-Type Neural Networks

Stochastic methods are commonly used in the training of neural networks in terms of associated weights or coefficients that have successfully performed better than traditional gradient-based techniques [19]. The literature shows a wide range of evolutionary design approaches either for architectures or for connection weights separately, in addition to efforts for them simultaneously [24] In the most GMDH type neural network, neurons in each layer are only connected to neuron in its adjacent layer as it was the case in Methods I and II previously reported in reference [20]. Taking this advantage, it was possible to present a simple encoding scheme for the genotype of each individual in the population as already proposed by authors [20]. The encoding schemes in generalized GMDH neural networks (GS-GMDH) must demonstrate the ability of representing different length and size of such neural networks.

In GS_GMDH neural network, Figure 1, neuron ad in the first hidden layer is connected to the output layer by directly going through the second hidden layer. Therefore, it is now very easy to notice that the name of output neuron (network's output) includes "ad" twice as abbcadad. In other words, a virtual neuron named adad has been constructed in the second hidden layer and used with abbc in the same layer to make the output neuron abbcadad as shown in the Figure 1. It should be noted that such repetition occurs whenever a neuron passes some adjacent hidden layers and connects to another neuron in the next 2^{nd}, or 3^{rd}, or 4^{th}, or … following hidden layer. In this encoding scheme, the number of repetition of that neuron depends on the number of passed hidden layers, ñ, and is calculated as $2^{\tilde{n}}$. It is easy to realize that a chromosome such as abab bcbc, unlike chromosome abab acbc for example, is not a valid one in GS-GMDH networks and has to be simply re-written as abbc.

2.3. Genetic Operators for Generalized GMDH Network Reproduction

The genetic operators of crossover and mutation can now be implemented to produce two offspring from two parents. The natural roulette wheel selection method is used for choosing two parents producing two offspring. The crossover operator for two selected individuals is simply accomplished by exchanging the tails of two chromosomes from a randomly chosen point as shown in Figure 2. It should be noted, however, such a point could only be chosen randomly from the set 2^1, 2^2,…, 2^{n_l+1}, where n_l is the number of hidden layers of the chromosome with the smaller length.

It is very evident from Figures 2 and 3 that the crossover operation can certainly exchange the building blocks information of such generalized GMDH neural networks so that the two types of generalized GMDH and conventional GMDH type neural networks can be converted to each other, as can be seen from Figure 3.

In addition, such crossover operation can also produce different length of chromosomes that, in turn, lead to different sizes of either generalized GMDH type or conventional GMDH type network structures. Similarly, the mutation operation can contribute effectively to the diversity of the population. This operation is simply accomplished by changing one or more symbolic digits as genes in a chromosome to other possible symbols, for example, abbcadad to abbccdad. It is very evident that mutation operation can also convert a generalized GMDH type network to a conventional GMDH type network or vice versa. It should be noted that such evolutionary operations are

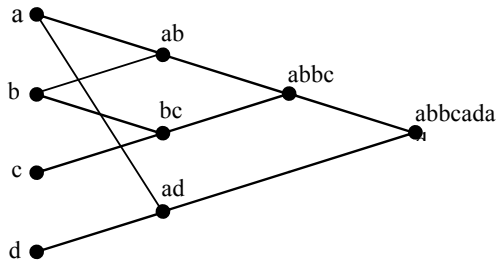

Figure 1. A generalized GMDH network structure of a chromosome.

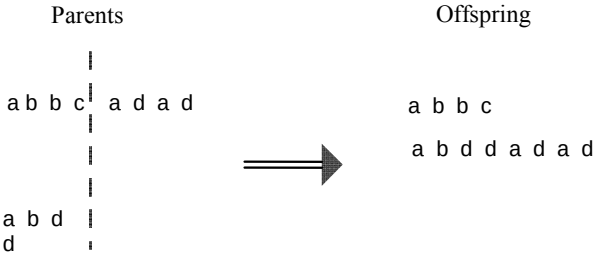

Figure 2. Crossover operation for two individuals in generalized GMDH network.

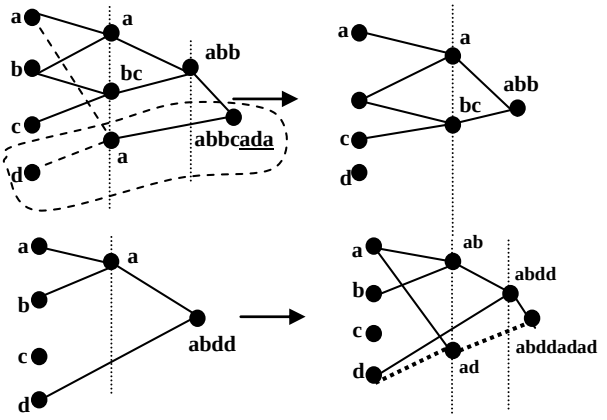

Figure 3. Crossover operation on two generalized GMDH networks.

acceptable provided a valid chromosome is produced. Otherwise, these operations are simply repeated until a valid chromosome is constructed.

The incorporation of genetic algorithm into the design of such GMDH-type neural networks starts by representing each network as a string of concatenated sub-strings of alphabetical digits. The fitness, (Φ), of each entire string of symbolic digits which represents a GMDH-type neural network to model explosive cutting process is evaluated in the form:

$$\Phi = 1/E, \tag{14}$$

Where E, is the mean square of error given by Equation 6, is minimized through the evolutionary process by maximizing the fitness Φ. The evolutionary process starts by randomly generating an initial population of symbolic strings, each as a candidate solution. Then, using the aforementioned genetic operations of roulette wheel selection (crossover and mutation), the entire populations of symbolic strings to improve gradually. In this way, GMDH type neural network models with progressively increasing fitness, Φ are produced until no further significant improvement is achievable.

3. SIMULATION OF STALL CONTROL

3.1. Governing Equations and Sgs Modeling

In LES models, the small eddies are separated by filters from large eddies that contain most of the energy. There are two key characteristics in LES approach. One is the filter function; the other is the SGS model. First, the flow variables are decomposed into large-scale components (denoted by an overbar) and small subgrid scale components by employing a filtering operation. The resulting equations thus govern only the dynamics of large eddies and these large-scale variables can be defined by the filtering operation:

$$\bar{f}(x) = \frac{1}{\Delta V}\int f(x')dx', \tag{15}$$

Where ΔV is the control volume. The above is a simple top-hat filter in physical space that is applied implicitly by the finite-volume discretization.

Applying the filtering operation to the Navier-Stokes equations and assuming that the filtering and differentiation operations commute, the following equations for the evolution of the large-scale motions are obtained:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \tag{16}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j}, \tag{17}$$

Where i (= 1, 2, 3) indicates the spatial dimension; $\bar{p}$ is the filtered pressure; $\bar{u}_i(x_i,t)$ is the resolved velocity field. The incompressible form of the subgrid scale stress is:

$$\tau = \overline{u_i u_j} - \bar{u}_i \bar{u}_j, \quad (18)$$

The subgrid scale stresses resulting from the filtering operation are unknown and require modeling. Majority of the subgrid scale models in use today are eddy viscosity models. The models assume proportionality between the anisotropic part of the SGS stress tensor $\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk}$ and the resolved scale strain rate tensor $\bar{S}_{ij}$ as Equation 19:

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2\nu_t \bar{S}_{ij}, \quad (19)$$

Where μ_t is the subgrid scale turbulent viscosity, and $\bar{S}_{ij}$ is defined by

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right), \quad (20)$$

3.2. RNG-Based Subgrid-Scale Model in LES The most basic of subgrid scale models is proposed by Smagorinsky and further developed by Smagorinsky [54] and Lilly [55]. In the Smagorinsky-Lilly model, the eddy viscosity is simulated by

$$\nu_t = (C_s \Delta)^2 |\bar{S}|, \quad (21)$$

$$|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}, \quad (22)$$

Where $\bar{\Delta}$ is the filter length and the model parameter C_s should be optimized according to a type of turbulent flow field ($0.1 \le C_s \le 0.23$). The near-wall correction is also required, and the Van Driest type wall damping function, $f_{VD} = 1-\exp(-y^+/25)$, is traditionally introduced as $\bar{\Delta} \to \bar{\Delta} f_{VD}$ ($y^+ = u_\tau y/\nu$ is the wall unit, y is the wall normal coordinate and u_τ is the wall friction velocity). However it was not used in the present investigation, mainly because this approach lacks generality as it involves the distance of a cell centroid to the wall which is difficult to evaluate in complex geometries and whereas this model is based on high Reynolds number flow, predicts non-zero turbulence viscosity in laminar flows. Instead, it was decided to use a RNG-based model that predict more accurately transition or relaminarization and have the correct near-wall behavior as opposed to the constant-coefficient Smagorinsky model. Yakhot, et al [56] have obtained an RNG subgrid scale stress model by performing recursive elimination of infinitesimal bands of small scales. In this RNG-based SGS model, the effective viscosity, $\mu_{eff} = \mu + \mu_t$, is given by:

$$\mu_{eff} = \mu\,[1 + H(x)]^{1/3}, \quad (23)$$

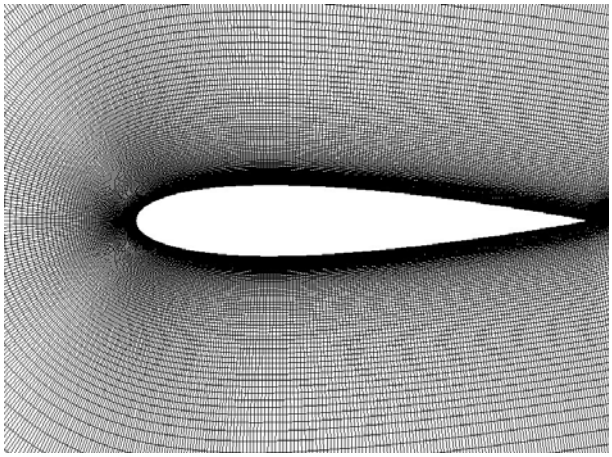

Figure 4. A portion of the C-type grid for NACA0015.

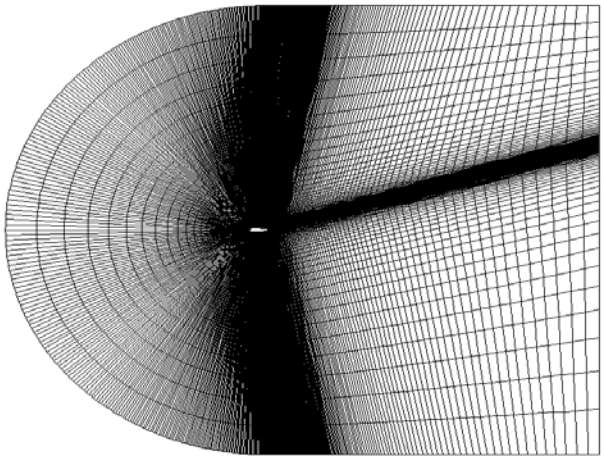

Figure 5. Schematic representation of computational grid.

Where H(x) is the ramp function defined by

$$H(x)=\begin{cases}x, x\geq 0,\\ 0, x<0\end{cases} \quad (24)$$

Here, the variable x equals to $(\mu_s\mu_{eff}/\mu^3)-C$, where $\mu_s=(C_{RNG}\Delta V^{1/3})^2\sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$. Based on the RNG theory, the constants C_{RNG} and C are given by 0.157 and 100, respectively [56].

In highly turbulent regions, the filtering operation results in very high subgrid viscosity compared to the molecular viscosity, $\mu_s\geq\mu$ and $\mu_{eff}\cong\mu_s$. In this limit, the RNG theory based subgrid scale model returns to the Smagorinsky model with a different model constant. In weakly turbulent regions, the argument of the Heaviside function is negative, and the effective viscosity is equal to the molecular viscosity. The RNG SGS model in this limit correctly yields zero SGS viscosity in low Reynolds number flows without any ad-hoc modifications to model the low-Reynolds-number effects encountered in near-wall region.

3.3. Numerical Scheme The governing Navier-Stokes equations are integrated by finite volume method for a structured grid system. All the flow variables are stored at geometric centers of the arbitrary shaped cells. In this study, the time-dependent terms were handled through an implicit second-order backward differencing in time. The transport equations were discretized using the finite-volume method. LES model is sensitive to the spatial discretization errors because eddies near the cut-off wave number is still energetic [57]. It has been reported that the truncation error overwhelms the contribution of SGS force for upwind and upwind-biased schemes [58]. So the central differencing scheme was used to approximate the convective terms at the faces of the control volumes for LES model. The PISO algorithm [59] is used to deal with the pressure-velocity coupling between the momentum and the continuity equations. In order to avoid the generation of a check-board pressure field, the velocity interpolation method at cell faces proposed by Rhie and Chow [60] is employed.

3.4. Grid Setup A structured C-type grid system is used for calculation of the flow field around the NACA0015 airfoil.

This grid is clustered close to the wall and in the near-wake, in order to get an accurate description of the vortex shedding. Figure 4 shows an overview of the C-mesh used around the airfoil. The computational domain is plotted in Figure 5. The upstream boundary is 12 chord lengths away from the leading edge of the airfoil while the outlet boundary is placed 20 chord lengths downstream of the trailing edge.

On the outer boundary, the uniform flow boundary conditions are imposed at the upstream boundary and the right (outflow) boundary condition is set to a zero velocity gradient condition. A no-slip wall boundary condition is taken on the airfoil surface.

To model the synthetic jet actuator, suction/ blowing type boundary condition is used. Thus, a prescribed velocity distribution is imposed at the jet boundary:

$$\vec{U}=A\sin(\omega t)f(s)\vec{d}_{jet} \quad (25)$$

Where the amplitude $A=\sqrt{\frac{C}{2H}U_\infty\sqrt{\langle C_\mu\rangle}}$, the prescribed frequency of oscillation $\omega=(F^+U_\infty 2\pi/C)$ and $\vec{d}_{jet}$ is a vector of unit length representing the direction of the jet outlet. α_{jet} is the angle between $\vec{d}_{jet}$ and the wall. f(s) is the distribution of the velocity along the jet boundary. It is supposed to have a negligible influence on the flow, as shown by Donovan, et al [9]. Therefore, a ‘‘top hat’’ distribution is adopted, corresponding to f(s) = 1. F^+ is the non-dimensional frequency and the oscillatory momentum blowing coefficient $< C_\mu >$ is defined as:

$$< C_\mu >= 2(H/c)(< u_{jet} >/U_\infty)^2 \quad (26)$$

This choice of boundary conditions coupled with the use of a relatively large computational domain ensures that the results are relatively insensitive to the domain size. This is demonstrated conclusively by re-computing the flow with a different domain size. All parameters chosen in the computation are dimensionless. A special attempt was made to

ensure that the near-wall y^+ values of the airfoil surface were kept within 0.5.

The near flowfield around the airfoil is the most sensitive computation areas; hence, the number of grid points in these areas is most critical. To test for grid independence, three sets of grids, with increasing grid density (labeled 1, 2, and 3), are studied, and their results are listed in Table 1.

The rows PS, SS, Wake, Ny and Nb pts correspond to the number of cells on the pressure side, on the suction side, in the wake, in the direction normal to the wall and the total number of cells, respectively. These grids are studied under a Reynolds number of 12.7×10^6, and computational results for different angles of attack are compared in Table 2 and Figure 6. The differences in the computational results between set 1 and set 2, and between set 2 and set 3, are less than 3 %. To maintain grid-resolution consistency at different locations and relatively high grid resolution at the jet and also for economy of computations, the relatively dense grid of set 2 is adopted in the current computation.

3.5. Parameters Selection and Verification Work In the current investigation, three parameters

TABLE 1. Characteristics of the Meshes Used.

MESH	PS	SS	WAKE	NY	NB PTS
M1	200	200	48	130	644.80
M2	245	262	96	155	109.556
M3	292	368	144	180	175140

TABLE 2. Coarse and Dense Grid C_l and C_d Comparison.

α	$1C_l$	$2C_l$	$3C_l$	$1C_d$	$2C_d$	$3C_d$
0	0.001043	0.001228	0.001048	0.013833	0.013943	0.013625
6	0.820219	0.82245	0.824278	0.02754	0.027585	0.027528
10	1.259591	1.260957	1.267991	0.042659	0.043681	0.044585
12	1.373413	1.395179	1.382324	0.063401	0.065946	0.067107
14	1.225949	1.261897	1.236193	0.12167	0.126616	0.12851
16	1.056942	1.0828	1.123305	0.245	0.235354	0.236623

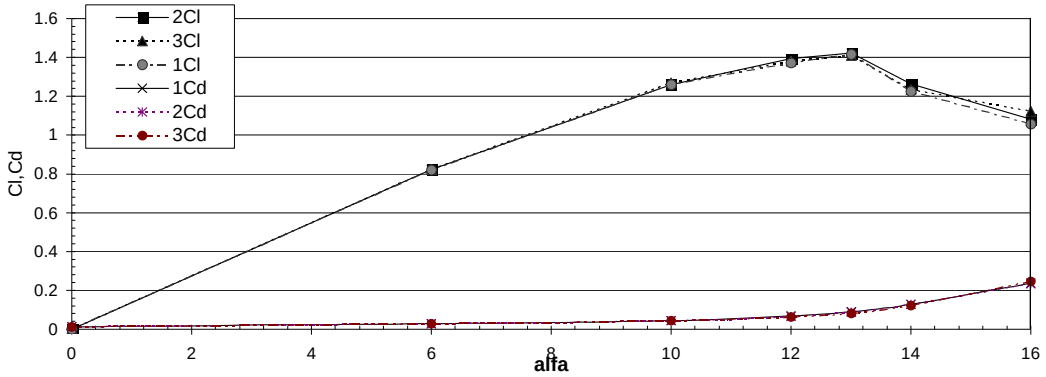

Figure 6. Grid independence study of the grids in Table 1.

of the synthetic jet are selected to optimize, called design variables, namely, jet momentum coefficient C_μ, non-dimensional jet frequency F^+ and direction of the outlet α_{jet}. In our numerical investigation, the range of jet momentum coefficient is selected to be from 0.005 to 0.1. It has been proposed [61] that a jet momentum coefficient C_μ around 0.002 is necessary to have some impact on the flow pattern. The dimensionless frequency F^+ is changed within 0.7 to 4 based on previous researches [63,64], which showed that $F^+ > 4$ will not increase lift significantly. The jet angle α_{jet} is changed from 0 deg to 60 deg with respect to the wall. This rang covers more of the airfoil jet angle than those used in previous experimental and numerical studies. All cases are under Reynolds number 12.7×10^6 and angle-of-attack 18-deg conditions.

The numerical solution is validated first by comparing with experimental data [62] without jet actuator. The value of parameter which were used in the simulations were chosen to match the specifications of the experiments of flow separation control performed by Seifert [62] at $Re = 12.7 \times 10^6$ by a synthetic jet located at $S_{jet} = 10$ % of the chord L from the leading edge, with a slot width h = 0.2 % of the chord.

In order to reduce the computational time required to converge to a solution with blowing, the baseline case solution (without excitation) was used as the initial flow conditions for the blowing computations.

Figure 7 shows the computed lift coefficient (C_l) versus angle of attack compared with the experimental data. The computed result is reasonably close to the graph obtained experimentally. The stall angle is overestimated by 1° and the maximum C_l by 7 %. This is so because, in general, there exists a difficulty for numerical approaches to match the lift coefficient for angles of attack above the separation angle [65]. The general trend is, though, correctly captured. Given in Figure 8 is the comparison of the distributions of pressure coefficient on the airfoil between numerical and experimental data, which shows that the LES approach is capable of prediction the airfoil flow at $\alpha = 14°$. The computed controlled airfoil surface pressure distributions are compared with the experimental data for $F^+ = 2$, $< C_\mu > = 0.03$ % and $\alpha = 14°$ in Figure 9 where the good agreement is achieved at this angle of attack.

The numerically obtained lift coefficient C_l for controlled case with variation in angle of attack (α)

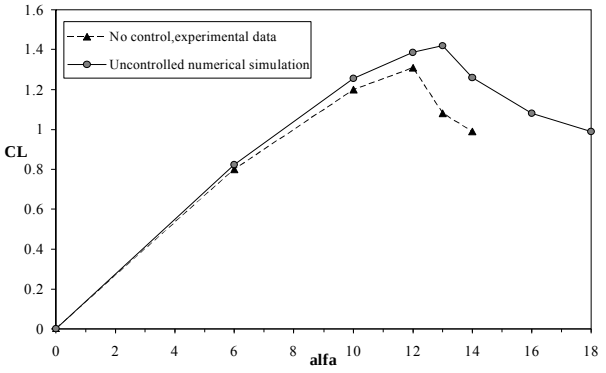

Figure 7. Baseline lift coefficient for NACA0015 airfoil computed with experiments.

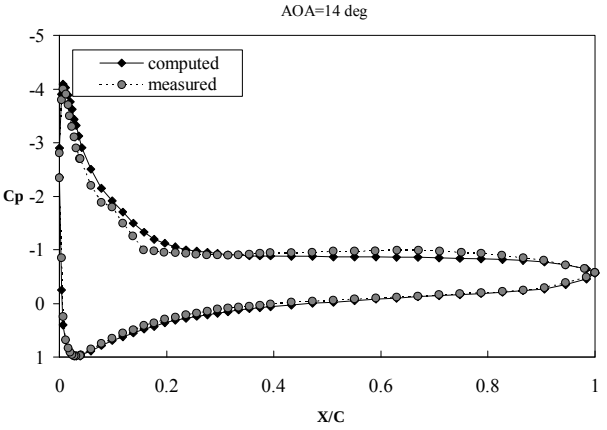

Figure 8. Pressure distributions on NACA0015 airfoil for the case of zero actuation.

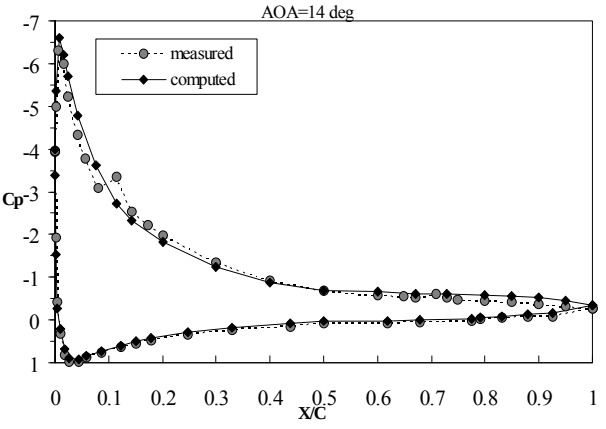

Figure 9. Pressure distributions on NACA0015 airfoil for controlled case.

are compared with experimental data in Figure 10. It can be seen from Figure 10 that the numerical results closely follow the experimental data. Despite this, a small over-prediction of the lift coefficient can be seen.

Figure 7 and 10 also clearly show the benefits of the oscillatory excitation through the increase in the mean lift coefficient of the controlled case with respect to the lift coefficient for the uncontrolled one. As seen, for an incidence of 8˚ and 10˚ the synthetic jet has no significant influence on the lift coefficient. However the slope of the curve is maintained till 13˚ thanks to the actuation, whereas the slope begins to decrease after 10˚ for the baseline airfoil. In the controlled case, the lift decrease occurs at about 15˚, delaying the stall by 2˚.The maximum lift is increased of 16 %.

4. MODELING OF LIFT AND DRAG COEFFICIENT USING GMDH-TYPE NEURAL NETWORK

The input-output data pairs used in such modeling involve two different data tables obtained from numerical simulations discussed in section 3. The first table consist of three variables as inputs namely, momentum coefficient (C_μ), reduced frequency (F^+) and angle w.r.t. the wall (α_{jet}) and one output which is lift coefficient (C_l) for stall control on NACA0015 airfoil with a synthetic jet. The second table consists of the same three variables as inputs and another output which is drag coefficient (C_d). These tables consist of the total 144 pattern numbers which have been obtained from the numerical simulations to train GMDH-type neural networks. However, in order to demonstrate the prediction ability of the evolved GMDH-type neural networks, the data have been divided into two different sets, namely, training and testing sets. The training set, which consists of 116 out of 144 inputs-output data pairs, is used for training the neural network models using the method presented in section two. The testing set, which consists of 28 unforeseen input-output data samples during the training process, is merely used for testing to show the prediction ability of such evolved GMDH-type neural network models.

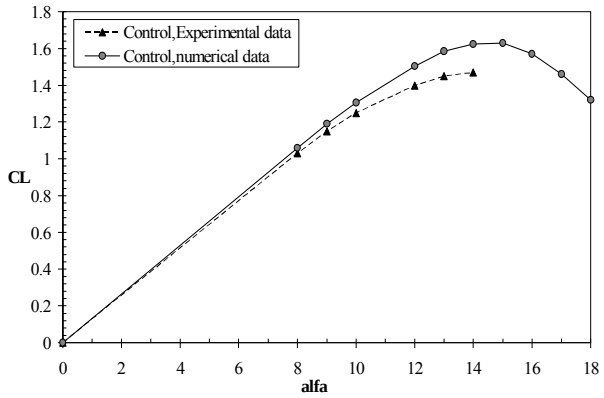

Figure 10. Comparison of numerical results versus experimental data regarding the effect of SJA actuation on the lift coefficient.

The GMDH-type neural networks are now used for such input-output data to find the polynomial model of lift and drag coefficient with respect to their effective input parameters. In order to genetically design such GMDH-type neural network described in previous section a population of 25 individuals with a crossover probability of 0.7 and mutation probability of 0.07 has been used in 200 generation that no further improvement has been achieved for such population size. The structure of the evolved 2-hidden layer GMDH-type neural networks are shown in Figures 11 and 12 corresponding to the genome representations of acbbaabc for lift coefficient and abbcaaac for drag coefficient which a,b and c stand for reduced frequency, momentum coefficient, and angle w.r.t. the wall, respectively. The corresponding polynomial representation of such model for lift coefficient is as follows:

$$Y_{13} = 1.6951 - 0.1394F^+ - 0.00427\alpha_{jet} - 0.0046(F^+)^2 + 0.000002\alpha_{jet}^2 + 0.0006F^+.\alpha_{jet} \quad (27a)$$

$$Y_{23} = 0.9691 + 14.7363C_\mu - 0.0021\alpha_{jet} - 98.52C_\mu^2 + 0.000016\alpha_{jet}^2 - 0.0427C_\mu.\alpha_{jet} \quad (27b)$$

$$Y_{2311} = -1.0511 + 2.2163Y_{23} + 0.5161F^+ - 0.39282Y_{23}^2 - 0.07057F^{+2} - 0.1961Y_{23}.F^+ \quad (27c)$$

$$Y_{2213} = -1.5589 + 2.4728C_\mu + 3.7376Y_{13} - 61.3852C_\mu^2 - 1.31069Y_{13}^2 + 5.4151C_\mu.Y_{13} \quad (27d)$$

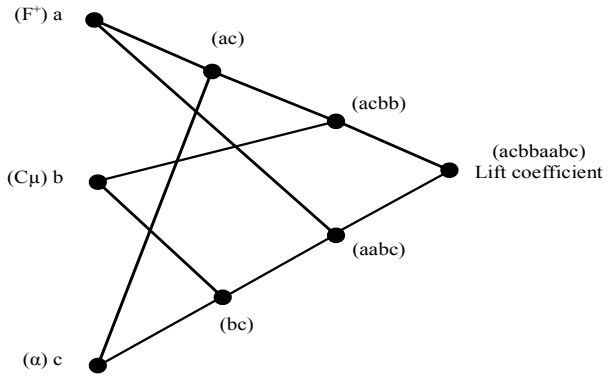

Figure 11. Evolved structure of generalized GMDH neural network for lift coefficient.

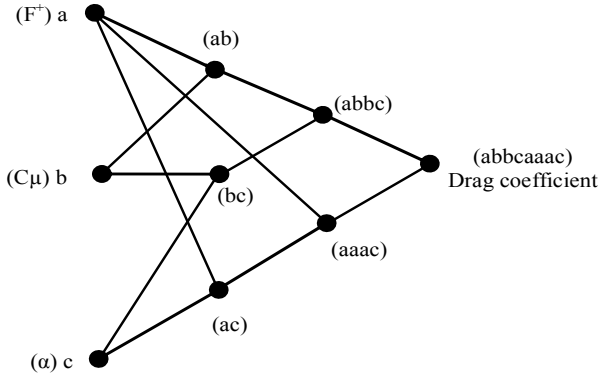

Figure 12. Evolved structure of generalized GMDH neural network for drag coefficient.

$$C_l = 0.1041 + 0.8213Y_{2311} + 0.0039Y_{2213} - 0.5201Y_{2311}^2 - 0.1331Y_{2213}^2 + 0.7238Y_{2311}.Y_{2213} \quad (27e)$$

Similarly, the corresponding polynomial representation of the model for drag coefficient is in the form of:

$$Y'_{13} = -0.16912 + 0.4517F^+ + 0.0034\alpha_{jet} - 0.0692(F^+)^2 - 0.000016(\alpha_{jet})^2 - 0.00037F^+.\alpha_{jet} \quad (28a)$$

$$Y'_{12} = 0.3637 + 0.2413F^+ - 10.2625C_\mu - 0.04104(F^+)^2 + 66.5225(C_\mu)^2 + 0.60108F^+.C_\mu \quad (28b)$$

$$Y'_{23} = (0.007 - 0.1399C_\mu + 0.00002\alpha_{jet} + 1.0519(C_\mu)^2 - 0.0000002(\alpha_{jet})^2 + 0.00034C_\mu.\alpha_{jet}).100 \quad (28c)$$

$$Y'_{1113} = 0.0575 + 1.4161F^+ - 6.1414Y'_{13} - 0.2362F^{+2} + 6.5884Y'^2_{13} - 0.0062F^+.Y'_{13} \quad (28d)$$

$$Y'_{1223} = -0.1921 - 0.4284Y'_{12} + 2.1788Y'_{23} + 3.2007Y'^2_{12} + 0.3974Y'^2_{23} - 4.2879Y'_{12}.Y'_{23} \quad (28e)$$

$$C_d = 0.2458 + 0.1089Y'_{1113} - 0.6467Y'_{1223} + 2.7647Y'^2_{1113} + 3.4302Y'^2_{1223} - 4.2635Y'_{1113}.Y'_{1223} \quad (28f)$$

The very good behavior of the GMDH-type neural network models is also depicted in Figures 13 and 14 for testing data of both lift and drag coefficient, respectively.

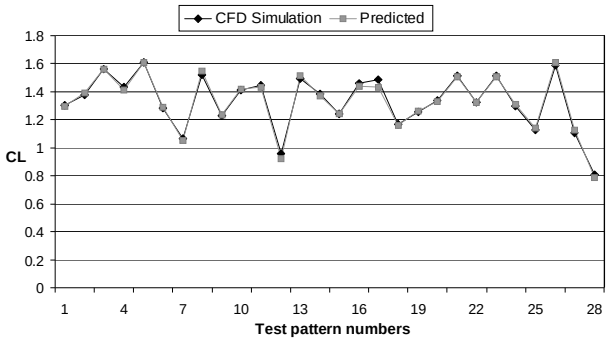

Figure 13. Variation of lift coefficient with input data.

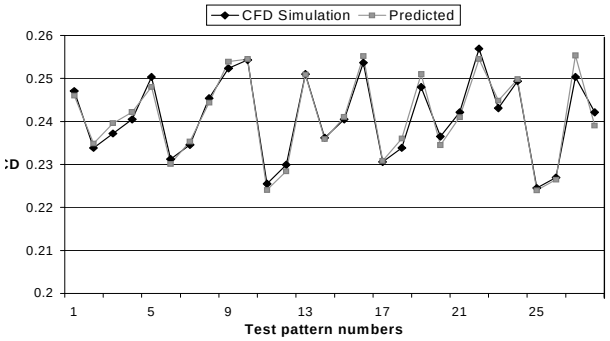

Figure 14. Variation of drag coefficient with input data.

It is clearly evidenced that the evolved GMDH-type neural network in terms of simple polynomial equations can successfully model and predict the output of testing data that has not been used during the training process.

The models obtained in this section can now be utilized for a Pareto multi-objective optimization of the control parameters considering lift (C_l) and drag (C_d) as conflicting objectives. Such study may unveil some interesting and important optimal design principles that would not have been obtained without the use of a multi-objective optimization approach.

5. MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization which is also called multi criteria optimization or vector optimization has been defined as finding a vector of decision variables satisfying constraints to give acceptable values to all objective functions [38,53]. In general, it can be mathematically defined as follows:

Find the vector $X^* = [x_1^*, x_2^*, ..., x_n^*]^T$ to optimize

$$F(X) = \left[f_1(X), f_2(X), ..., f_k(X)\right]^T \quad (29)$$

Subject to m inequality constraints:

$$g_i(X) \le 0, \quad i = 1 \text{ to } m \quad (30)$$

and p equality constraints:

$$h_j(X) = 0, \quad j = 1 \text{ to } p \quad (31)$$

Where $X^* \in \Re^n$ is the vector of decision or design variables, and $F(X) \in \Re^k$ is the vector of objective functions which each of them be either minimized or maximized. However, without loss of generality, it is assumed that all objective functions are to be minimized. Such multi-objective minimization based on Pareto approach can be conducted using some definitions.

5.1. Definition of Pareto Dominance A vector $U = \left[u_1, u_2, ..., u_k\right] \in \Re^k$ is dominance to vector $V = \left[v_1, v_2, ..., v_k\right] \in \Re^k$ (denoted by $U \prec V$) if and only if $\forall i \in \{1,2,...,k\}$, $u_i \le v_i \wedge \exists j \in \{1,2,...,k\}: u_j < v_j$. In other words, there is at least one u_j which is smaller than v_j whilst the remaining u 's are either smaller or equal to corresponding v 's.

5.2. Definition of Pareto Optimality A point $X^* \in \Omega$ (Ω is a feasible region in $\Re^n$ satisfying Equations 2 and 3) is said to be Pareto optimal (minimal) with respect to the all $X \in \Omega$ if and only if $F(X^*) < F(X)$. Alternatively, it can be readily restated as

$$\forall i \in \{1,2,...,k\}, \quad \forall X \in \Omega - \{X^*\} \quad f_i(X^*) \le f_i(X) \quad \wedge \quad \exists j \in \{1,2,...,k\}: f_j(X^*) < f_j(X).$$

In other words, the solution X^* is said to be Pareto optimal (minimal) if no other solution can be found to dominate X^* using the definition of Pareto dominance.

5.3. Definition of Pareto Set For a given MOP, a Pareto set $\mathcal{P}^*$ is a set in the decision variable space consisting of all the Pareto optimal vectors $\mathcal{P}^* = \{X \in \Omega \mid \nexists\, X' \in \Omega : F(X') < F(X)\}$. In other words, there is no other X' as a vector of decision variables in Ω that dominates any $X \in \mathcal{P}^*$.

5.4. Definition of Pareto Front For a given MOP, the Pareto front $\mathcal{PF}^*$ is a set of vector of objective functions that are obtained using the vectors of decision variables in the Pareto set $\mathcal{P}^*$, that is $\mathcal{PF}^* = \{F(X) = (f_1(X), f_2(X),..., f_k(X)): X \in \mathcal{P}^*\}$. In other words, the Pareto front $\mathcal{PF}^*$ is a set of the vectors of objective functions mapped from $\mathcal{P}^*$.

Evolutionary algorithms have been widely used for multi-objective optimization because of their natural properties suited for these types of problems. This is mostly because of their parallel or population based search approach. Therefore, most of the difficulties and deficiencies within the classical methods in solving multi-objective

optimization problems are eliminated. For example, there is no need for either several runs to find the Pareto front or quantification of the importance of each objective using numerical weights. In this way, the original non-dominated sorting procedure given by Goldberg [33] was the catalyst for several different versions of multi-objective optimization algorithms [36,37]. However, it is very important that the genetic diversity within the population be preserved sufficiently. This main issue in MOPs has been addressed by many related research works [49]. Consequently, the premature convergence of MOEAs is prevented and the solutions are directed and distributed along the true Pareto front if such genetic diversity is well provided. The Pareto based approach of NSGA-II [50] has been used recently in a wide area of engineering MOPs because of its simple yet efficient non-dominance ranking procedure in yielding different level of Pareto frontiers. However, the crowding approach in such state of the art MOEA [51] has been modified by the work of some of authors [29,30] which is used in this paper to obtain the Pareto front of two conflicting objective function, namely, C_L and Cd.

6. PARETO OPTIMIZATION OF STALL CONTROL USING POLYNOMIAL NEURAL NETWORK MODELS

In order to investigate the optimal control parameters of the synthetic jet in different condition of design variables, i.e. momentum coefficient, reduced frequency and the direction of the outlet, the polynomial neural network models obtained in previous sections are now deployed in a multi-objective optimization procedure. The two conflicting objective in this study are time-averaged lift coefficient (C_L) and time-averaged drag coefficient (C_D) to be simultaneously optimized with respect to the design variables, C_μ, F^+ and α_{jet}. Evidently, it can be observed that the C_L is maximized whilst the C_D is minimized in the set of objective functions (C_L,C_D). The evolutionary process of Pareto multi-objective optimization is accomplished by using the modified NSGA-II approach [29,30] where a population size of 100 has been chosen in different runs with crossover probability Pc and mutation probability Pm is 0.95 and 0.1, respectively. However, in order to refine the obtained results as non-dominated points, the Pareto front is moved to a better one by using a single-objective optimization technique for each non-dominated points. The scheme is represented in Figure 15, where each non-dominated points is regarded as an initial guess for the single-objective optimization technique by keeping one objective fixed whilst improving the other objective by changing the design vector. The corresponding Pareto front of two objectives C_L and C_D has been shown in Figure 16, using the hybridized approach of this work. It is clear from this figure that choosing appropriate value for the parameters of the synthetic jet for obtaining a better value of one objective would cause a worse value of another objective.

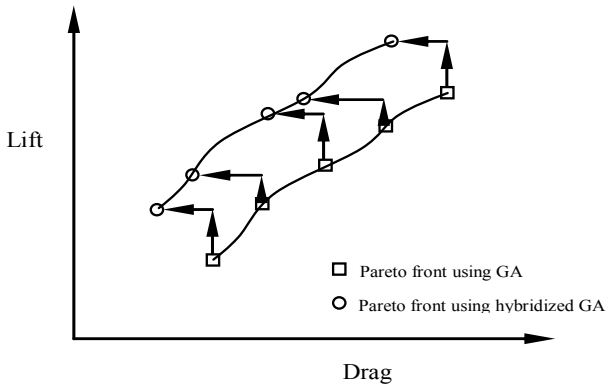

Figure 15. Hybridization procedure of initial Pareto front.

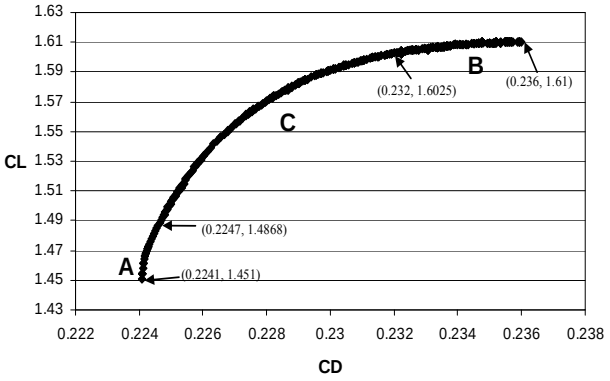

Figure 16. Pareto front of two objectives lift and drag.

However, if the set of decision variables is selected based on each of the Pareto sets, it will lead to the best possible combination of those two objectives. In other words, if any other pair of decision variables is chosen, the corresponding values of the pair of objectives, i.e. C_L and C_D, will locate a point inferior to the corresponding Pareto front. Such inferior area in the space of the two objectives is in fact bottom/right side of Figure 16. Clearly, there are some important optimal design facts between the two objective functions which have been discovered by the Pareto optimization of the polynomial neural network models obtained using the numerical data of stall control with a synthetic jet. Such important design facts would be very important to the designer to switch from one optimal solution to another for achieving different trade-off requirements of the objectives and could not have been found without the multi-objective Pareto optimization of those polynomial models. In this way, two sections, A and B, can be seen from Figure 16 which demonstrate these important optimal design facts. Section A exhibits decrease of lift coefficient (C_L) whilst the amount of decrease in drag coefficient is very small. Section B exhibits a significant increment of drag coefficient (C_d) with a small change in lift (C_L). Therefore, changing the parameters of a synthetic jet as decision variables should be in such a way that stall control performance on airfoil in terms of lift (C_L) and drag (C_d) lies between sections A and B (section C) of the Pareto optimal front. This will not only ensure the optimal process of the stall control but also prohibit such deficiency involved in both sections A and B.

In this numerical study, the end optimized best non-dominated individuals with the maximum control fitness for the angle of injection (α_{jet}) is: $\alpha_{jet} < 2^{\circ}$. These results are consistent with several reported numerical and experimental works that demonstrated the effective control angles on the airfoil for synthetic jets. For example, regarding the appropriate direction of injection of the fluid (α_{jet}), it was found in numerical experiments [66] that no effective flow separation control can be achieved for a jet exiting normally to the airfoil surface (α_{jet} = 90 deg). It was also found, on the other hand, that for α_{jet} = 30 deg large amplitude oscillations of the computed loads (e.g. lift and drag forces) are obtained. This load oscillation is undesirable and therefore a jet exit angle $\alpha_{jet} \leq 10$ deg is recommended by the numerical simulations. Other examples include Seifert, Darabi and Wygnanski [67], who studied the effect of periodic excitation at 0˚ (wall-jet) on a NACA0015; Gilarranz, et al [68] and Hassan, et al [69] who studied a synthetic jet that is almost tangential to the wall.

Figures 17 and 18 demonstrate the corresponding variations of lift coefficient (C_l) versus the other decision variables, namely momentum coefficient (C_μ) and reduced excitation frequency (F^+), respectively. It can be seen that the corresponding sections of A and B of the Pareto front (Figure 16) can be verified in these figures. Therefore, it can be readily concluded that section A corresponds to lower values of reduced excitation frequency (F^+) around 0.8 with some small values of momentum coefficient (C_μ) less than 0.059. On the other hand, section B corresponds to higher value of reduced

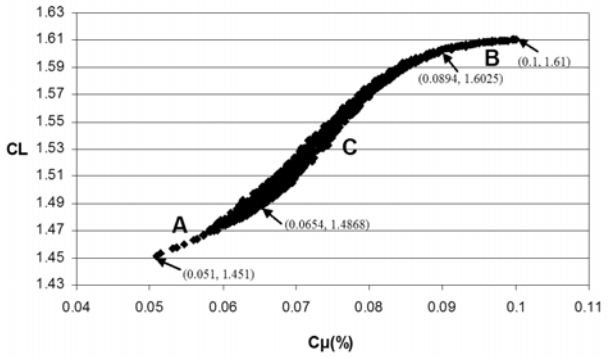

Figure 17. Optimal variation of the lift with respect to the momentum coefficient.

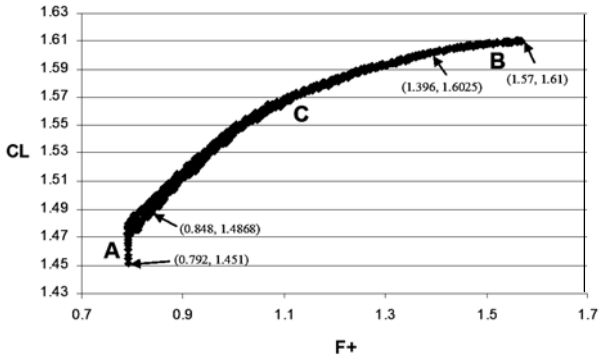

Figure 18. Optimal variation of the with respect to the reduced frequency.

frequency (F^+) greater than 1.44 with same large values of momentum coefficient (C_μ) greater than 0.09. The corresponding values of decision variables, (C_μ) and F^+, in conjunction with the Pareto front are shown in Figure 19. Evidently, choosing the value of reduced frequency $0.85 < (F^+) < 1.38$ with a range of momentum coefficient (C_μ), $0.065 < C_\mu < 0.085$, according to Figure 19, in conjunction with $\alpha_{jet} < 2^\circ$ ensures the best possible combination of stall control performance in terms of lift (C_l) and drag (C_d) relating to the area C of the Pareto optimal front shown in Figure 16. For example, it can be seen that for non-dominated optimum design point (1.58,0.229) relating to the area C of the pareto optimal front lift coefficient increase by 60 % with respect to the baseline airfoil and 20 % with respect to the initial control parameters. On the other hand, drag coefficient decrease by 31 % w.r.t the baseline and 13 % w.r.t the initial control parameters. Figure 20 demonstrate the corresponding variation of drag (C_d) versus the momentum coefficient (C_μ). The Pareto front obtained from the GMDH-type neural network model (Figure 16) has been superimposed with the corresponding numerical simulation results in Figure 21. It can be clearly seen that such obtained Pareto front lies on the best possible combination of the objective values of numerical data which demonstrates the effectiveness of the approach of this paper both in deriving the model and in obtaining the Pareto front.

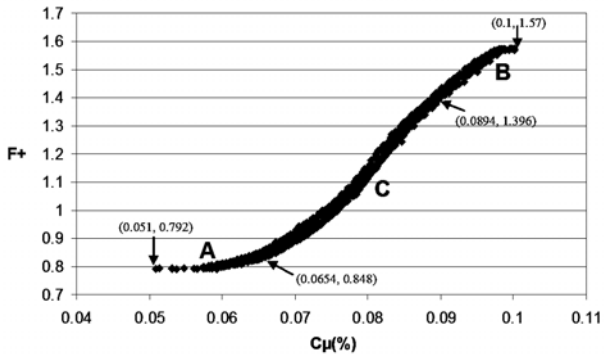

Figure 19. Optimal variation of reduced frequency with respect to the momentum coefficient.

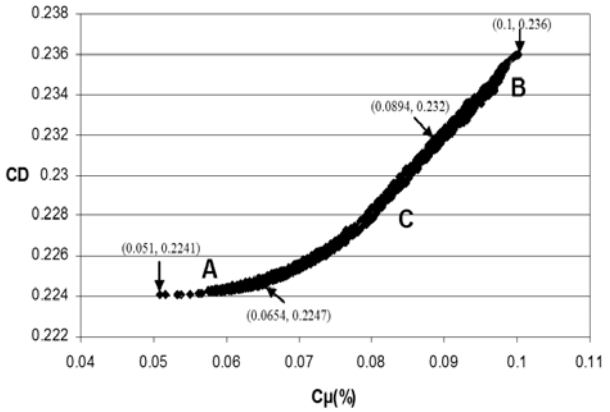

Figure 20. Optimal variation of the drag with respect to the momentum coefficient.

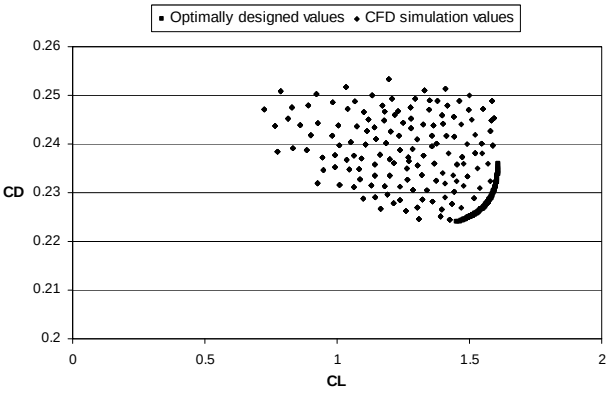

Figure 21. Overlay graph of the obtained optimal Pareto front with the numerical data.

7. CONCLUSION

Genetic algorithms have been successfully used both for optimal design of generalized GMDH type neural networks models of stall control using a synthetic jet and for multi-objective Pareto based optimization of such control process. The simulation of aerodynamic stall control using a synthetic jet actuator is firstly performed on a NACA0015 airfoil at Reynolds number of 12.7×10^6 and angle-of-attack 18-deg. the LES predictions for time-average lift versus angle of attack and pressure coefficient agreed quite well with the experimental data of [62]. Then, two different polynomial relations for lift and drag of controlled NACA0015 airfoil using a synthetic jet with different parameters or design variables have been found by evolved GS-GMDH type neural network using input-output data obtained from numerical simulation. The derived polynomial

models have been used in an evolutionary multi-objective Pareto based optimization process so that some interesting and informative optimum design aspects have been revealed for aerodynamic stall control with respects to the control parameters, i.e. momentum coefficient, reduced frequency and the direction of the outlet. The evolutionary multi-objective optimization process has helped to discover important relationships with relatively few efforts of modeling preparation that would otherwise have required at least a very through mathematical analysis. Such combined application of GMDH neural network modeling of numerical input-output data and subsequent non-dominated Pareto optimization process of the obtained models is very promising in discovering useful and interesting design relationships.

8. NOMENCLATURE

C	Airfoil Chord
C_L	Lift Coefficient
C_D	Drag Coefficient
C_P	Pressure Coefficient
R_e	Reynolds Number
α	Airfoil Angle of Attack
P	Pressure
U_∞	Free-Stream Velocity
h	Slot Width
μ_t	Subgrid-Scale Turbulent Viscosity
ρ	Density
F^+	Reduced Frequency
C_μ	Blowing Momentum Coefficient
x_i	Spatial Coordinate, i = 1, 2,…
u_i	Velocity Component
ν	Kinematic Viscosity
$\vec{d}_{jet}$	Jet Direction
$H(x)$	Ramp Function
τ	Subgrid Scale stress
$\overline{S}_{ij}$	Strain rate Tensor
$\overline{\Delta}$	Filter Length
Φ	Fitness
E	Mean Square of Error
X^*	Vector of Optimal Design Variables
$F(x)$	Vector of Objective Functions
P^*	Pareto Set (Set of Decision Variables)
PF^*	Pareto Front (Set of Objective Functions)

8.1. Subscripts

∞	Free-Stream Conditions
jet	Conditions at Blowing Slot

8.2. Superscripts

‘	Fluctuating Variables
–	Filtered Variables

9. REFERENCES

1. Hassan, A., Straub, F. and Charles, B. D., “Effects of Surface Blowing/Suction on the Aerodynamics of Helicopter Rotor Blade-Vortex Interactions-A Numerical Simulation”, ***J. Am Helicopter Soc***., Vol. 42, (1997), 182-194.
2. Seifert, A., Darabi, A. and Wygnanski, I., “Delay of Airfoil Stall by Periodic Excitation”, ***AIAA J***., Vol. 33, No. 4, (1996), 691-707.
3. Seifert, A., Bashar, T., Koss, D., Shepshelovich, M. and Wygnanski, I., “Oscillatory Blowing: A Tool to Do Delay Boundary Layer Separation”, ***AIAA J***., Vol. 31, No. 11, (1993), 2052-2060.
4. Smith, B. and Glezer, A., “Vectoring and Small-Scale Motions Effected in Free Shear Flows using Synthetic Jet Actuators”, ***AIAA Paper***, (1997), 0213-0221.
5. Gilarranz, J., Traub, L. and Rediniotis, O., “Characterization of a Compact, High Power Synthetic Jet Actuator for Flow Separation Control”, ***AIAA Paper***, (2002), 0127-0135.
6. Smith, D., Amitay, M., Kibens, V., Parekh, D. and Glezer, A., “Modification of Lifting Body Aerodynamics using Synthetic Jet Actuators”, ***AIAA Paper***, (1998), 0209-0218.
7. Wu, J., Lu, X., Denney, A., Fan, M. and Wu, J., “Post-Stall Lift Enhancement on an Airfoil by Local Unsteady Control”, ***AIAA Paper***, Lift, Drag and Pressure Characteristics, Part I, (1997), 2063-2071.
8. Wu, J., Lu, X., Wu, J., “Post-Stall Lift Enhancement on an Airfoil by Local Unsteady Control”, ***AIAA paper***, Mode Competition and Vortex Dynamics, Part II, (1997), 2064-2072.
9. Donovan, J. F., Kral, L. D. and Cary, A. W., “Active Control Applied to an Airfoil”, ***AIAA Paper***, (1998), 0210-0218.
10. Ekaterinaris, J., “Active Flow Control of Wing Separated Flow”, ***ASME FEDSM'03 Joint Fluids Engineering Conference***, Honolulu, Hawai, U.S.A., (July 6-10, 2003).
11. Astrom, K. J. and Eykhoff, P., “System Identification, A Survey”, ***Automatica J***., Vol. 7, (1971), 123-162.
12. Sanchez, E., Shibata, T. and Zadeh, L. A., “Genetic Algorithms and Fuzzy Logic Systems”, ***World Scientific***, Riveredge, New Jersey, U.S.A., (1997).
13. Kristinson, K. and Dumont, G., “System Identification and Control using Genetic Algorithms”, ***J. IEEE Trans Syst Man Cybern***, Vol. 22, No. 5, (1992), 1033-1046.

14. Koza, J., "Genetic Programming, on the Programming of Computers by Means of Natural Selection", ***MIT Press***, Cambridge, MA, U.S.A., (1992).
15. Iba, H., Kuita, T., Degaris, H. and Sator, T., "System Identification using Structured Genetic Algorithms", ***Proc. of 5th Int. Conf. on Genetic Algorithms***, ICGA'93, U.S.A., (1993).
16. Rodríguez-Vázquez, K., "Multi-Objective Evolutionary Algorithms in Non-Linear System Identification", PhD Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K., (1999).
17. Fonseca, C. M. and Fleming, P. J., "Nonlinear System Identification with Multi-Objective Genetic Algorithms", ***Proceedings of the 13th World Congress of the International Federation of Automatic Control***, Pergamon Press, San Francisco, California, U.S.A., (1996), 187-192.
18. Liu, G. P. and Kadirkamanathan, V., "Multi-Objective Criteria for Neural Network Structure Selection and Identification of Nonlinear Systems using Genetic Algorithms", ***IEEE Proceedings on Control Theory and Applications***, Vol. 146, No. 5, (1999), 373-382.
19. Ivakhnenko, A. G., "Polynomial Theory of Complex Systems", ***IEEE Trans Syst Man Cybern***, SMC-1, (1971), 364-378.
20. Farlow, S. J., "Self-Organizing Method in Modelling: GMDH Type Algorithm", Marcel Dekker Inc., Paris, French, (1984).
21. Mueller, J. A. and Lemke, F., "Self-Organising Data Mining: An Intelligent Approach to Extract Knowledge From Data", Pub. Libri, Hamburg, Germany, (2000).
22. Iba, H., Degaris, H. and Sato, T., "A Numerical Approach to Genetic Programming for System Identification", ***J. Evolutionary Computation***, Vol. 3, No. 4, (1996), 417-452.
23. Nariman-Zadeh, N., Darvizeh, A. and Ahmad-Zadeh, G. R., "Hybrid Genetic Design of GMDH-Type Neural Networks Using Singular Value Decomposition for Modelling and Prediction of the Explosive Cutting Process", Proceedings of the I MECH E Part B, ***Journal of Engineering Manufacture***, Vol. 217, (2003), 779-790.
24. Nariman-Zadeh, N., Darvizeh, A., Felezi, M. E. and Gharababaei, H., "Polynomial Modelling of Explosive Compaction Process of Metallic Powders using GMDH-Type Neural Networks and Singular Value Decomposition", ***Journal of Modelling and Simulation in Materials Science and Engineering***, Vol. 10, No. 6, (2002), 727-744.
25. Nariman-Zadeh, N., Darvizeh, A., Darvizeh, M. and Gharababaei, H., "Modelling of Explosive Cutting Process of Plates using GMDH-Type Neural Network and Singular Value Decomposition", ***Journal of Materials Processing Technology***, Vol. 128, No. 1, (2002), 80-87.
26. Porto, V. W., "Evolutionary Computation Approaches to Solving Problems in Neural Computation", Handbook of Evolutionary Computation Back, T., Fogel, D. B. and Michalewicz, Z., Editors. Institute of Physics Publishing and New York, Oxford University Press, (1997), D1.2:1-D1.2:6.
27. Yao, X., "Evolving Artificial Neural Networks", ***Proceedings of IEEE***, Vol. 87, No. 9, (1999), 1423-1447.
28. Vasechkina, E. F. and Yarin, V. D., "Evolv. Ing Polynomial Neural Network by Means of Genetic Algorithm: Some Application Examples", ***J. Complexity International***, Vol. 9, (2001), 729-744.
29. Nariman-Zadeh, N., Atashkari, K., Jamali, A., Pilechi, A. and Yao, X., "Invers Thermodynamic Pareto Optimization of Turbojet Engines using Multi-Objective Genetic Algorithms", ***J. of Engineering Optimization***, Vol. 37, 2005, 437-462.
30. Atashkari, K., Nariman-Zadeh, N., Gölcü, M., Khalkhali A. and Jamali, A., "Modelling and Multi-Objective Optimization of a Variable Valve-Timing Spark-Ignition Engine using Polynomial Neural Networks and Evolutionary Algorithms", ***J. of Energy Conversion and Management***, Vol. 48, No. 3, (2006), 1029-1041.
31. Arora, J. S., "Introduction to Optimum Design", Mcgraw-Hill, New York, U.S.A., (1989).
32. Rao, S. S., "Engineering Optimization: Theory and Practice", John Wiley and Sons, New York, U.S.A., (1996).
33. Goldberg, D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, New York, U.S.A., (1989).
34. Back, T., Fogel, D. B. and Michalewicz, Z., "Handbook of Evolutionary Computation", Institute of Physics Publishing and Oxford University Press, New York, U.S.A., (1997).
35. Renner, G. and Ekart, A., "Genetic Algorithms in Computer Aided Design", ***J. Computer-Aided Design***, Vol. 35, (2003), 709-726.
36. Srinivas, N. and Deb, K., "Multi-Objective Optimization using Non-Dominated Sorting in Genetic Algorithms", ***J. Evolutionary Computation***, Vol. 2, No. 3, (1994), 221-248.
37. Fonseca, C. M. and Fleming, P. J., "Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization", ***Proc. of the Fifth Int. Conf. on Genetic Algorithms***, Forrest, S., Editor, San Mateo, CA, Morgan Kaufmann, (1993), 416-423.
38. Coello Coello, C. A. and Christiansen, A. D., "Multi-Objective Optimization of Trusses using Genetic Algorithms", ***J. Computers and Structures***, Vol. 75, (2000), 647-660.
39. Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers, New York, U.S.A., (2002).
40. Pareto, V., "Cours D'economic Ploitique", Rouge, Lausanne, Switzerland, (1896).
41. Rosenberg, R. S., "Simulation of Genetic Populations with Biochemical Properties", PhD Thesis, University of Michigan, Ann Harbor, Michigan, U.S.A., (1967).
42. Schaffer, J. D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", ***Proc. of First Int. Conf. on Genetic Algorithms and Their Applications***, Grefenstette, J.J., Editor, Lawrence Erlbaum, London, U.K., (1985), 93-100.
43. Zitzler, E., Thiele, L., "An Evolutionary Algorithm for

Multi-Objective Optimization: The Strength Pareto Approach", Tech. Report 43, Computer Engineering and Communication Network Lab, Swiss Federal Ins. of Tech., Zurich, Switzerland, (1998).
44. Knowles, J. and Corne, D., "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multi-Objective Optimization", ***Proc. of the 1999 Congress on Evolutionary Computation***, Piscataway, IEEE Service Center, New Jersey, U.S.A., (1999), 98-105.
45. Horn, J., Nafpliotis, N. and Goldberg, D. E., "A Niched Pareto Genetic Algorithm For Multi-Objective Optimization", ***Proceedings of the First IEEE Conference on Evolutionary Computation***, IEEE World Congress on Computational Intelligence, Piscataway, IEEE Service Centre, New Jersey, U.S.A., Vol. 1, (1994), 82-87.
46. Coello Coello, C. A., "A Comprehensive Survey of Evolutionary Based Multi-Objective Optimization Techniques", Knowledge and Information Systems, ***An Int. Journal***, Vol. 3, (1999), 269-308.
47. Deb., K., "Multi-Objective Optimization Using Evolutionary Algorithms", John Wiley, U.K., (2001).
48. Khare, V., Yao, X. and Deb, K. "Performance Scaling of Multi-Objective Evolutionary Algorithms", ***Proc. of Second International Conference on Evolutionary Multi-Criterion Optimization***, (EMO'03), Portugal, (2003).
49. Toffolo, A. and Benini, E., "Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms", ***Evolutionary Computation***, MIT Press, Vol. 11, No. 2, (2003), 151-167.
50. Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", ***J. IEEE Trans. on Evolutionary Computation***, Vol. 6, No. 2, (2002), 182-197.
51. Coello Coello, C. A. and Becerra, R. L., "Evolutionary Multi-Objective Optimization using A Cultural Algorithm", ***IEEE Swarm Intelligence Symp***., U.S.A., (2003), 6-13.
52. Sarker, R., Liang, K. H. and Newton, C., "A New Continuous Optimization Multi-Objective Evolutionary Algorithm", ***European Journal of Operational Research***, Vol. 140, (2002), 12-23.
53. Osyezka, A., "Multicriteria Optimization for Engineering Design", Gero, J. S., Editor, Design Optimization, Academic Press, New Jersey, U.S.A., (1985), 193-227.
54. Smagorinsky, J., "General Circulation Experiments with the Primitive Equations, I. The Basic Experiment", ***J. Mon. Weather Rev***., Vol. 91, (1963), 99-164.
55. Lilly, D. K., "The Representation of Small Scale Turbulence in Numerical Simulation Experiments", ***Proc. IBM Scientific Computing Symposium on Environmental Sciences***, Goldstine, H. H., Editor., IBM Form, No. 320-1951, (1967), 195-210.
56. Yakhot, A., Orszag, S. A., Yakhot, V. and Israeli, M., "Renormalization Group Formulation of Large-Eddy Simulation", ***Journal of Scientific Computing***, Vol. 4, (1989), 139-158.
57. Park, N., Yoo, J. Y. and Choi, H., "Discretization Errors in Large Eddy Simulation: on the Suitability of Centered and Upwind-Biased Compact Difference Schemes", ***Journal of Computational Physics***, Vol. 198, (2004), 580-616.
58. Mittal, R. and Moin, P., "Suitability of Upwind-Biased Finite-Difference Schemes for Large-Eddy Simulation of Turbulent Flows", ***AIAA Journal***, Vol. 35, (1997), 1415-1418.
59. Issa, R. I., "Solution of The Implicitly Discretized Fluid Flow Equations by Operator Splitting", ***J. Comput Phys***., Vol. 62, (1986), 40-65.
60. Rhie, C. and Chow, W., "Numerical Study of Turbulent Flow Past an Airfoil with Trailing Edge Separation", ***AIAA J***., Vol. 21, (1983), 1525-32.
61. Gilarranz, J. L. and Rediniotis, O. K., "Compact, High-Power Synthetic Jet Actuators for Flow Separation Control," ***AIAA Paper***, (2001), 0737-0745.
62. Seifert, A. and Latunia, G. P., "Oscillatory Excitation of Unsteady Compressible Flows Over Airfoils at Flight Reynolds Numbers", ***37th AIAA Aerospace Sciences Meeting and Exhibit***, Reno, NV, U.S.A., (January 11-14, 1999).
63. Greenblatt, D., Darabi, A., Nishri, B. and Wygnanski, I., "Some Factors Affecting Stall Control with Particular Emphasis on Dynamic Stall", ***30th AIAA Fluid Dynamics Conference, AIAA Paper***, Norfolk, VA, U.S.A., (June 28-July 1, 1999), 3504-3512.
64. Zhou, M. D., Fernholz, H. H., Ma, H. Y., Wu, J. Z. and Wu, J. M., "Vortex Capture by a Two-Dimensional Airfoil with a Small Oscillating Leading-Edge Flap", ***AIAA Paper***, (1993), 3266-3574.
65. Anderson, W. K., Thomas, J. L. and Rumsey, C. L., "Application of Thin-Layer Navier-Stokes Equations Near Maximum Lift", ***AIAA Paper***, (1984), 0049-0057.
66. Ekaterinaris, J. A., "Numerical Investigations of Dynamic Stall Active Control for Incompressible and Compressible Flows", ***AIAA Paper***, (2000), 4333-4341.
67. Seifert, A., Darabi, A. and Wygnanski, I., "Delay of Airfoil Stall by Periodic Excitation", ***AIAA J. Aircraft***, Vol. 33, No. 4, (1996), 691-698.
68. Gilarranz, J. L. and Rediniotis, O. K., "Compact, High-Power Synthetic Jet Actuators for Flow Separation Control", ***AIAA Paper***, (2001), 0737-0745.
69. Hassan, A. and Janakiram, R. D., "Effects of Zero-Mass Synthetic Jets on the Aerodynamics of the NACA-0012 Airfoil", ***Journal of the American Helicopter Society***, Vol. 43, No. 4, (October 1998), 2326-2337.