Proceedings of the 38th
Conference on Decision & Control
Phoenix, Arizona USA • December 1999

Algorithms for Shear Flow Control and Optimization

Petros Koumoutsakos
Institut für Fluiddynamik, ETH Zürich
and
Center for Turbulence Research, NASA Ames
petros@eniac.ethz.ch

Parviz Moin
Center for Turbulence Research, NASA Ames/Stanford University
moin@ctr.stanford.edu

Abstract

We present an outline of our research efforts in the area of flow control and optimization. We focus on identifying low order models that can be implemented in active control strategies using (i) our understanding of fundamental physical processes such as vorticity generation and (ii) machine learning and optimization algorithms such as neural networks and evolution strategies.

The results presented herein encompass a wide variety of problems, such as drag minimization, neural net modeling of the near wall structures, enhanced jet mixing and parameter optimization in turbine blade film cooling.

1 Introduction

Flow control has been a fundamental concept in fluid mechanics research in this century. In the early 1900's research was focused on the development of experimental procedures that would elucidate the governing flow phenomena in order to devise efficient control devices. A number of empirical methods were proposed, such as rotating cylindrical sails for ship propulsion and the placement of wires around wing profiles (a precursor of riblets) for drag minimization. On the second half of the century developments such as the discovery of coherent structures in the wake of bluff body flows and the understanding of the processes of flow separation led to a number of devices (e.g vortex generators, splitter plates, mass transpiration, etc) for the efficient manipulation of flow structures in experiments and realistic engineering configurations. In the 80's and 90's the advent of Direct Numerical Simulations (DNS) provided us with a thorough understanding of fundamental processes, such as the mechanisms of skin friction drag in turbulent flows [8]. DNS of turbulent flows have been used as the testing grounds for a number of control algorithms, such as the opposition control scheme[5], feedback control using models derived via POD[14] or neural networks [12], the feedback control of vorticity generation [9],[10] and optimal [3] and suboptimal control [13] strategies. These simulations have provided us with valuable insight into the behavior of controlled flows. Moreover these algorithms have demonstrated that the effective control of turbulence in engineering applications requires strategically placed, micro/nano devices that would be capable of sensing and actuating frequencies in the order of a few MHz.

Recent progress in manufacturing has provided us with an unprecedented array of such potent devices, such as MEMS [7] capable of sensing and effecting the detailed structure of a turbulent flow. However, the proper integration of control devices and control algorithms in realistic applications remains a challenging problem. While a wealth of different possible geometry modifications and/or open loop actions have been proposed, little effort has focused in trying to devise a concise optimization algorithm to adjust the diverse control parameters. The development of low order models that can be used in conjunction with control theories is a challenging problem that, when solved, could lead to drastically improved designs.

In this article we outline two methodologies for developing low order models. In Section 1 we outline the vorticity flux feedback control algorithm which has resulted in drag reduction of up to 40% in DNS of turbulent channel flows. A key aspect of this approach is the development of an explicit formulation relating the actuator strengths to the wall vorticity flux. Such relations can be devised for simple idealized actuation but are not readily available for realistic sensors and actuators. We envision neural network approaches as an effective way of developing such models and incorporating them in feedback control algorithms. In section 2 we present some preliminary results from the application of neural networks as a method to construct low order models, describing the near wall dynamics in turbulent flows. Neural networks are viewed as a general procedure of model formation encompassing schemes such as the Proper Orthogonal Decomposition.

Another key issue in the effort to reduce time to market of new engineering designs is the optimization of the design parameters in an efficient manner. The design cycle usually involves multi-objective and multidisciplinary optimization problems, requiring the iterative solution of empirical formulas, the appropriate integration of numerical simulations and the incorporation of physical understanding of the various aspects of the problem. At the same time, the optimization cycle of the physical problem must take into consideration financial and manufacturing constraints. In flow related problems, this optimization cycle has benefited from advances in optimization theory which usually aim at tackling the most costly aspects of the optimization problem, such as the solution of the Navier-Stokes equations. Powerful techniques such as the adjoint procedure have been implemented successfully in the design cycle of aircrafts [19].

However, such optimization strategies are usually based on the efficient calculation of gradients of functions relating the quantity to be optimized to the parameters of the problem. Such gradients are not always readily available as often the optimization cycle would involve empirical formulas and cost functions that are difficult to express analytically in terms of the optimization problem. Moreover, gradient based algorithms are usually converging to local extrema. Therefore, the result strongly depends on the initial selection of parameters.

Evolution strategies [18] are optimization techniques that avoid the problems associated with the use of gradients as they require only the calculation of the cost function at each point in the parameter space. They are operating based on natural principles of evolution such as mutation, recombination and selection. These operations are adapted so that the algorithm automatically develops and attempts to optimize a *model landscape* relating the cost function to its parameters. Compared with gradient based techniques, their convergence rate is usually much lower, thus requiring large numbers of iterations that could be unrealistic for some problems of engineering interest. On the other hand, they are highly parallel algorithms that efficiently exploit todays powerful parallel computer architectures and they are more likely than gradient based algorithms to identify a global optimum. This latter aspect makes them attractive in many engineering applications where the fitness landscape cannot be assumed unimodal. In section 3 we report results from the application of evolutionary algorithms to a number of diverse areas such as drag minimization in cylinder flows, jet mixing optimization and turbine blade film cooling.

2 Vorticity flux control

Vorticity is a fundamental quantity in fluid mechanics, as every quantity in the flow can be described by its evolution. In incompressible flows, vorticity is created only at the solid boundaries of the flow. Vorticity creation is quantified by the wall vorticity flux which can be calculated from the wall pressure gradients. The amount of vorticity created by certain types of idealized actuators, such as mass transpiration in DNS calculations, can be computed using a fractional step algorithm [10]).

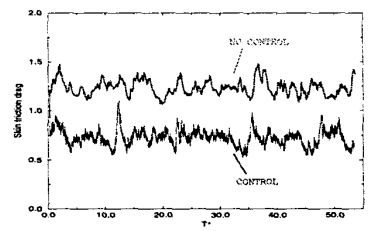

Figure 1: Vorticity flux control. Evolution of skin friction drag coefficient at $Re_\tau = 200$

For example for a two-dimensional flow over a flat wall, with a system of sources/sinks of strength q_j that are distributed uniformly over a panel of size d_j, centered at locations $x'_j, j = 1, 2, 3, ...N$ the induced vorticity flux at a point x_i on the wall is :

$$\nu\, \delta t\, \frac{\partial \omega}{\partial y}(x_i) = \sum_{j=1}^{N} \frac{q_j}{2\pi} \int_{-d_j/2}^{d_j/2} \frac{ds}{x - s} \qquad (1)$$

where $x = x_i - x'_j$. The methodology can be easily extended for a variety of actuators, such as wall acceleration, deformation, etc.

2.1 An active control strategy.

For the purposes of our control scheme we consider a series of vorticity flux (or equivalently pressure gradient) sensors on the wall at locations $x_i, i = 1, 2, 3, ...M$. We can explicitly determine the actuator strengths necessary to achieve a desired vorticity flux profile at the wall at a time instant, k, by solving the linear set of equations :

$$B\, u_k + X_{k-1} = D_k \qquad (2)$$

where $D_k = (\frac{\partial \omega^k}{\partial y}(x_1), \frac{\partial \omega^k}{\partial y}(x_2), ..., \frac{\partial \omega^k}{\partial y}(x_M))$ is an $M \times 1$ vector of the *desired* vorticity flux at the sensor locations, $X_{k-1} = (\frac{\partial \omega^{k-1}}{\partial y}(x_1), \frac{\partial \omega^{k-1}}{\partial y}(x_2), ..., \frac{\partial \omega^{k-1}}{\partial y}(x_M))$ is an $M \times 1$ vector of the *measured* vorticity flux at the sensor locations and $u_k = (q_1^k(x'_1), q_2^k(x'_2), ..., q_N^k(x'_N))$ is an $N \times 1$ vector of source strengths at the actuator locations, B is an $M \times N$ matrix whose elements B_{ij} are determined by evaluating the integrals in Eq.1.

The present technique gives us the flexibility to adapt the actuator strengths to specific constraints. Practical considerations may constrain the control to jet-like actuators, $q_j \geq 0, j = 1, ..., N$ or to a blowing and suction configuration with a net zero mass flux. Such constraints may be easily incorporated in the above scheme by appropriately adjusting matrix B. A square, invertible matrix is always possible by accordingly modifying the number and placement of sensors and actuators.

Results from the application of the vorticity flux control algorithm to turbulent channel flow have been reported in [10] for a low Reynolds number turbulent channel flow ($Re_\tau = 200$). A collocated arrangement of sensors and actuators was considered. In this arrangement the rows of sensors and actuators are located at alternating streamwise grid locations on the bottom wall. In Fig. 1 we present the drag coefficient for the uncontrolled and the controlled turbulent channel flow. These results indicate a drag decrease of up to 40% using out-of-phase control of the spanwise vorticity flux.

3 Neural networks as near wall flow models

A key aspect of the vorticity flux algorithm is the formulation of an explicit relation between the values of vorticity flux and actuator strengths. While this can be computed for idealized actuations, the generalization to realistic devices is not straightforward. We envision that such correlations can be produced using neural networks. In order to assess the validity of this approach we examine the reconstruction of the near wall structures using wall only information with the aid of linear and nonlinear neural nets.

3.1 The POD as a linear neural network

A *model reduction* can be accomplished by projecting the model equations, i.e. the Navier-Stokes equations, on a properly selected lower dimensional phase subspace. A reasonable choice for a "proper" selection criterion for the base of this manifold is the maximization of the energy content of the projection. This can be done by applying the Karhunen-Loeve decomposition to a data set that is representative of the dynamics of the system that we wish to approximate. This operation is called *Proper Orthogonal Decomposition* (POD)[2], or *Linear Principal Components Analysis* (PCA). The linear POD is an approximation of the flow vector v by a finite expansion of orthonormal functions ϕ_n such that : $v = V + \sum_{i=1}^{n} a_n(t)\phi_n(x)$. where V is the time averaged flow, ϕ_n is the set of the first n eigenvectors of the covariance matrix $C = E[(v_i - V)(v_j - V)]$; when this representation for v is substituted in the Navier Stokes equations, the original PDE model is transformed in an ODE model, composed by n equations.

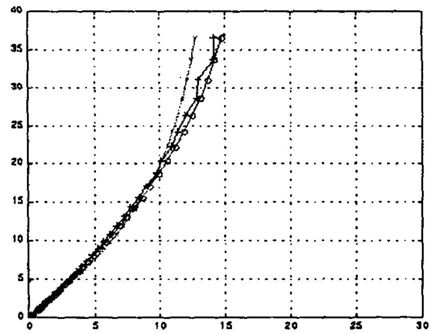

Figure 2: Streamwise and spanwise averaged profile. Blue: original, red: reconstruction using linear POD, black: reconstruction using NN

The POD can be expressed as a multi-layer feed-forward neural network. Such a network is defined by the number of layers, the specification of the output function for the neurons in each layer, and the weight matrices for each layer. Baldi and Hornik [1] have shown that training a linear neural network structure to perform an identity mapping on a set of vectors is equivalent to obtaining the POD of this set of vectors. A neural network performing the linear POD can be specified as a 2 layer linear network:

$$x = W_1 v; \quad \hat{v} = W_2 x$$

where $\hat{v}$ is the reconstructed field, v is the original flow field, having N components, x is the reduced order representation of the field, having n components, and W_1 and W_2 are the network weight matrices, of sizes $N \cdot n$ and $n \cdot N$ respectively. Non-linearity can be introduced by a simple extension to this basic network[20]:

$$x = W_2 tanh(W_1 v); \quad \hat{v} = W_4 tanh(W_3 x)$$

This corresponds to a neural network model with 4 layers: the first one, with and $m \cdot N$ weight matrix W_1, nonlinear; the second one, with $n \cdot m$ weight matrix W_2, linear; the third one, also nonlinear, with $m \cdot n$ weight matrix W_3, and the last one, linear with $N \cdot m$ weight matrix W_4. However the resulting system of ODEs is more involved as compared to the one resulting from the application of the linear POD.

We have conduceted extensive [16] comparisons of POD and nonlinear NN for the stochastically forced Burger's equation (a classical 1D model for turbulent flows [4]) and for the reconstruction of the near wall structures using wall only information. In the latter case, making a Taylor series expansion of the velocity field for small distances above the wall one can reconstruct a second order model of the velocity field using wall only information, such as the shear stresses and the pressure at

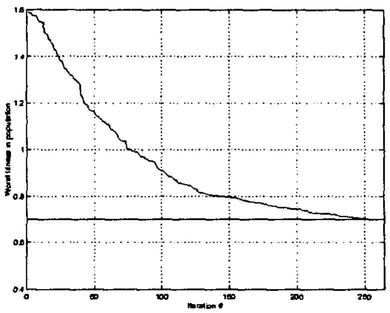

Figure 3: Worst fitness/drag in the population as a function of the optimization iterations. The horizontal line is the desired target drag coefficient

the wall. For the streamwise velocity field we have that :

$$u(\mathbf{x},t) = \omega_z^w y + \frac{Re}{2}\frac{\partial P^w}{\partial x}y^2 + \mathcal{O}(y^3) \qquad (3)$$

where $\omega_z^w = S^w$ are the shear stresses, P^w is the wall pressure. This model has been found to be accurate only up to $y^+ = 10$ for a wide range or Reynolds numbers. It can be improved by using a neural network to approximate the higher order terms as functions of wall quantities:

$$u(\mathbf{x},t) = \omega_z^w y + \frac{Re}{2}\frac{\partial P^w}{\partial x}y^2 + \mathbf{M}(P^w, S^w) \qquad (4)$$

where $\mathbf{M}(P^w, S^w)$ denotes a linear (i.e. a POD) or nonlinear neural network model using as input information the wall pressure and shear stresses. This model has been applied to a turbulent channel flow with $Re_\tau = 250$, based on the channel half height. In Fig. 3 we present a snapshot of the streamwise and spanwise averaged u^+ profile, comparing a POD reconstruction and a nonlinear model reconstruction, using 2 neurons in the inner layer, 1280 inputs containing the shear stresses and wall pressures measured on the bottom of a minimal flow unit, and 49920 output neurons carrying the estimated velocities. The snapshot in Fig.2 results from a simulation using samples that the neural network did not use in the training phase, thus showing the good generalization of this model.

4 Evolutionary Optimization Algorithms

Neural networks are machine learning algorithms that attempt to generate low order models from existing data in an automatic fashion. Other constituents of machine learning algorithms are evolution strategies and genetic algorithms. Evolution strategies (ES) and Genetic Algorithms (GA) operate on a population with a number of individuals, each of them represented by a real or binary vector. For an optimization problem with n parameters, each vector comprises n elements. Three operators are defined to modify the population members: *I. Recombination/crossover*, that generates new trial solution points (offsprings), using some elements drawn from the population; *II. Mutation*, that randomly changes some of the offsprings components; *III. Selection*, that chooses the population elements that will be used by the crossover. For each population element a *fitness function* is defined, measuring in a quantitative way how close a given solution is to the desired goal. Based on their fitness, the old population members are compared with the newly generated ones, and the solutions with the better fitness constitute the new population members. In this way, iterating the selection-crossover-mutation process, the population evolves towards the optimal solution.

We have developed self-organizing genetic algorithms particularly suitable for finding clusters of good solutions [15]- a desirable scheme when seeking non-sharp, non-single point optima. A variable mutation operator, depending on the local fitness value and on the global success history of the population, allows the population to avoid local optima. The algorithm operates in a hierarchical fashion, by identifying well correlated clusters of population, leading to optimization schemes employing few strategically placed actuators.

In our research efforts we have been concerned with the convergence rate of evolution strategies. A crucial parameter is the adaptation of the step size of the evolution strategy as in effect this reflects the properties of the environment to the parameter population. A powerful control scheme for step size adaptation is the covariance matrix adaptation (CMA)[6],[17]. With this method, the step sizes are adapted using prior information. The adaptation of the mutation distribution with the CMA is independent of the coordinate system and – in combination with the adaptation of the global step size – yields a high convergence rate.

A further speed-up is achieved by combining the CMA-ES with an intermediate recombination that averages the variable vector elements of some of the parents. This combined method is called $(\mu/\mu_I, \lambda)$-CMA-ES where μ/μ_I denotes the recombination of μ_I out of μ parents.

In the $(\mu/\mu, \lambda)$-CMA-ES, the parameter vectors $\mathbf{x}_k^{(k+1)}, k = 1, ..., \lambda$ in generation $g+1$ are computed by

$$\mathbf{x}_k^{(g+1)} = \langle\mathbf{x}\rangle_\mu^{(g)} + \delta^{(g)}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\mathbf{z}_k, \qquad (5)$$

where the center of mass of the selected individuals is given by

$$\langle\mathbf{x}\rangle_\mu^{(g)} = \frac{1}{\mu}\Sigma_{j=1}^{\mu}\mathbf{x}_j^{(g)}. \qquad (6)$$

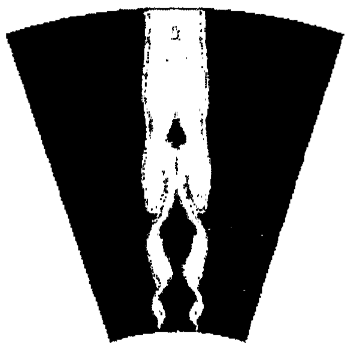

Figure 4: Contours of scalar concentration in jet flow at Re = 1500.No Control

Figure 5: Controlled flow - Bifurcating Jet

B and **D** are computed from the covariance matrix **C** such that the eigenvectors of **C** are the columns of **B** and the square roots of the eigenvalues of **C** are the elements of the diagonal matrix **D**:

$$\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)}(\mathbf{B}^{(g)}\mathbf{D}^{(g)})^{\mathrm{T}}. \quad (7)$$

The covariance matrix is formed from the evolution path, **s**, as :

$$\begin{aligned} \mathbf{s}^{(g+1)} &= (1-c)\cdot\mathbf{s}^{(g)} + c_u \cdot \tfrac{\sqrt{\mu}}{\delta^{(g)}}\left(\langle\mathbf{x}\rangle_\mu^{(g+1)} - \langle\mathbf{x}\rangle_\mu^{(g)}\right) \\ \mathbf{C}^{(g+1)} &= (1-c_{cov})\cdot\mathbf{C}^{(g)} + c_{cov}\cdot\mathbf{s}^{(g+1)}(\mathbf{s}^{(g+1)})^{\mathrm{T}}, \end{aligned}$$

where $1/c$ represents the accumulation time for the evolution path, and where $1/c_{cov}$ represents the averaging time for the covariance matrix. The accumulation time parameter, c, can be written in normalized form as $c_u = \sqrt{c(2-c)}$.

5 Optimization Results

These optimization algorithms have been applied to a number of optimization problems with sufficient success. The self-organizing genetic algorithm was applied in order to optimize the parameters of a prototypical cylindrical configuration [15]. In this configuration the surface of the cylinder is subdivided into 16 equal segments that are allowed to move independently tangentially to the surface of the body. Using a hierarchical clustering approach the evolutionary algorithm was able to identify automatically the critical points of the flow (near the uncontrolled flow separation points) while resulting in 50% drag reduction for two-dimensional flow at Re=250. In Fig.3 we show the convergence of the algorithm as a function of the number of iterations.

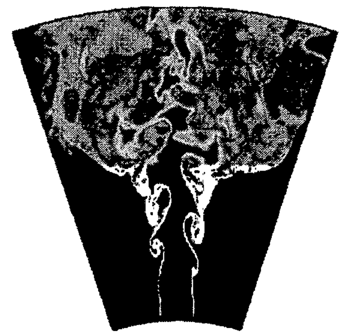

Figure 6: Controlled Flow - Flapping Jet

The CMA-ES strategy was applied to the parameter optimization of scalar mixing in DNS of jet flow at Re=1500 [11]. This is a challenging problem for evolution strategies as each iteration requires lengthy computations. Using various cost functions it was possible to identify previously unknown effective parameters that induce various types of behavior to the jet (Figs. 4,5,6).

Finally the CAM-ES algorithm was implemented in the realistic design cycle of a gas turbine blade film cooling [17]. This optimization cycle involved the optimization of parameters used in empirical algebraic formulas along with a heat conduction simulation program for the film cooling problem. The optimization goals were to reduce the coolant mass flow and at the same time achieve a homogeneous surface temperature while observing certain constraints in the range of the temperature distribution. Starting from a number of initial configurations, the optimization algorithm was always able to produce a number of improved designs, improving initial engineering estimates up to 25%. It should be emphasized that besides classical designs requiring large numbers of cooling rows in the leading edge of the blade the optimization algorithm was also able to identify novel designs involving certain ratios of number of rows on the two sides of the blade (Fig.7).

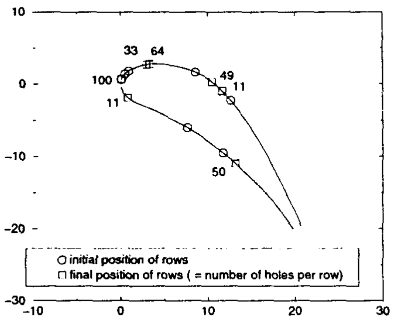

Figure 7: Initial and final position of the rows

6 Conclusions

We have presented results from the development and implementation of modeling and optimization algorithms based on the understanding of the fundamental physical phenomena and to the development and implementation of machine learning algorithms.

Our near future research in the area of flow control is directed in :

- development of low order models using information that is available not only in DNS but also in experimental and realistic engineering configurations. For example low order models that exploit information, such as pressure and shear stresses that can be obtained by wall mounted sensors.
- the integration of these low order models in feedback control algorithms and the optimization of their parameters using deterministic and stochastic optimization strategies

We envision that the synergy of scientists in fluid mechanics, control theory, artificial intelligence and manufacturing will be able to produce efficient control strategies capable of providing a leap in the performance of future engineering configurations.

References

[1] Baldi P., Hornik K., "Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima", Neural Networks, vol. 2, (1989)

[2] Berkooz G., Holmes P., and Lumley J., *The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows*, Annu Rev Fluid Mech, Vol. 25, 539-575, (1993).

[3] Bewley T.R. Optimal and robust control and estimation of transition and turbulence. *Ph.D. dissertation, Stanford University.*,(1999)

[4] Chambers D.H., Adrian R., Moin P.,Stewart D., Sung H.J., "Karhunen-Loeve expansion of Burgers' model of turbulence", Phys. Fluids, vol. 31, (1988)

[5] Choi H., Moin P. and Kim J.*Active turbulence control for drag reduction in wall bounded flows*, J. Fluid Mech., **262**,pp75-110, (1994)

[6] Hansen N., Ostermeier A., "Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES," *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pp. 650-654, (1997).

[7] Ho, C. & Tai, Y. Review: MEMS and its applications for flow control. J. of Fluids Engin. **118**, 437-447, (1996)

[8] Kim, J., Moin, P., & Moser, R. Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177**, 133-166., (1987)

[9] Koumoutsakos P. *Active Control of Vortex-Wall Interactions*, Phys. Fluids A, **9:12**, (1997).

[10] Koumoutsakos P. *Vorticity flux control for a turbulent channel flow*, Phys. Fluids , **11:2**, (1999).

[11] Koumoutsakos P., Müller S., Hilgers A. and Freund J., *Evolutionary Optimization of jet mixing parameters*, Phys. Fluids , (submitted).

[12] Lee C., Kim J., Babckock D. amd Goodman R., *Application of neural networks to turbulence control for drag reduction*, Phys. Fluids A, **9** 1740-1747,(1997)

[13] Lee, C., Kim, J. and Choi, H. *Suboptimal control of turbulent channel flow for drag reduction.* J. Fluid Mech. 358, 245-258, (1998)

[14] Lumley J. and Blossey P. *Control of Turbulence* Annu Rev Fluid Mech , Vol. 30, pp. 311-327, (1998)

[15] Milano M. and Koumoutsakos P. *A Self-Organizing genetic algorithm for cylinder drag minimization*, Phys. Fluids , (submitted).

[16] Milano M. and Koumoutsakos P. *Near wall turbulent flow reconstruction using neural networks*, IEEE Transactions of Neural Networks , (submitted).

[17] Müller S. and Koumoutsakos P. *Evolution Strategies for Film Cooling Optimizationg*, AIAA J. , (submitted).

[18] Rechenberg, I. Evolutionsstrategie : Optimierung technischer System nach Prinzipien der biologischen Evolution, (1973) Fromman-Holzboog, Stuttgart.

[19] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. L., Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," Journal of Aircraft, Vol. 36, No. 1, 1999.

[20] Takane Y., "Multivariate analysis by neural network models", Proceedings of the 63rd Annual Meeting of the Japan Statistical Society, 1995