Transactions of the Institute of Measurement and Control 31, 1 (2009) pp. 97–112

Seven tuning schemes for an ADALINE model to predict floor pressures in a subsonic cavity flow

Mehmet Önder Efe[1], Marco Debiasi[2], Peng Yan[3], Hitay Özbay[4] and Mohammad Samimy[5]

[1]Department of Electrical and Electronics Engineering, TOBB Economics and Technology University, Sögütözü Ankara, Turkey
[2]National University of Singapore, Temasek Laboratories, Singapore
[3]Enterprise Servo Engineering, Seagate Technology, Shakopee, MN 55379, USA
[4]Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800, Ankara, Turkey
[5]Department of Mechanical Engineering, The Ohio State University, Columbus, OH 43210, USA

This paper presents a simple yet effective one-step-ahead predictor based on an adaptive linear element (ADALINE). Several tuning schemes are studied to see whether the obtained model is consistent. The process under investigation is a subsonic cavity flow system. The experimental data obtained from the system is post-processed to obtain a useful predictor. The contribution of the paper is to demonstrate that despite the spectral richness of the observed data, a simple model with various tuning schemes can help to a satisfactory extent. Seven algorithms are studied, including the least mean squares (LMS), recursive least squares (RLS), modified Kaczmarz's algorithm (MK), stochastic approximation algorithm (SA), gradient descent (GD), Levenberg–Marquardt optimization technique (LM) and sliding mode-based tuning (SM). The model and its properties are discussed comparatively.

Key words: ADALINE; prediction; subsonic cavity flows.

Address for correspondence: Mehmet Önder Efe, TOBB Economics and Technology University, Department of Electrical and Electronics Engineering Sögütözü Cad. No. 43, Sögütözü, TR-06560 Ankara, Turkey. E-mail: onderefe@ieee.org

10.1177/0142331208090964

1. Introduction

Modelling efforts for aerodynamic flows have primarily the goal of developing a good understanding about the dynamics of the process. One approach for modelling is to exploit the physics of the problem (Rowley *et al*., 2001). Individual sub-models are proposed to represent the shear layer, scattering, cavity acoustics and receptivity in the form of parameterized transfer functions, which are tuned to match the real process dynamics (Yuan *et al*., 2004). Another major research direction focuses on proper orthogonal decomposition (POD) for modelling of shallow cavity flows (Yuan *et al*., 2005), based on modal analysis with numerical observations. A detailed review of the flow-induced cavity oscillations is presented in Cattafesta *et al*. (2003), where these major directions are emphasized.

This paper describes a third alternative for modelling based on input/output observations read from critical spatial locations. This makes it possible to address the difficulties stemming from the unavailability of the dynamical models of pre- and post-filtering devices, actuators, sensors and so on. It is therefore a good alternative to work with numerical data (measurements) containing the contribution of every component of the overall flow physics implicitly. This, however, makes it obligatory to merge the process dynamics and sensing and actuation periphery in a manner in which the modelling and control goals become achievable. In our earlier studies, neural networks (NNs) (Efe *et al*., 2005a) and fuzzy logic (Efe *et al*., 2005b) are shown to be useful for flow modelling purposes, whereas the approaches considered in this paper yield a finely tuned representative dynamic model, which has a significantly simpler structure than those obtained through the use of techniques mentioned so far.

Research on neural models has been carried out in the past decade to explore the usefulness of NN techniques in flow control reporting various degrees of success in each work. Among these are efforts exclusively focused on the numerical simulation of the flow model and of the corresponding control. Jacobson and Reynolds (1993) conducted a numerical study on the control of wall shear stress in a boundary layer by using feedforward NNs as controllers, which showed skin friction reduction by about 8%. Applications of generalized and specialized learning architectures are presented with the goal of inverting the plant dynamics. The neurocomputing techniques exploited in Jacobson and Reynolds (1993) have their roots in the pioneering work of Narendra and Parthasarathy (1990), and relevant applications are seen later on in Agarwal (1997). The study of active laminar flow control by Fan *et al*. (1993) showed that a properly trained NN can establish complex non-linear relationships between multiple inputs and outputs that are peculiar to an active flow control system. They also used experimental data but did not validate the control system experimentally. The work demonstrates the cancellation of wave disturbances in transitional boundary layers by a pre-trained NN. Sensors measure either wall pressure or wall shear stress. Training strategies and performance measures are considered, and fault tolerance capability of NN is emphasized. Faller *et al*. (1994) obtained an NN model of a pitching aerofoil based on experimental data. With limited training data, the model predicts unsteady surface pressure topologies within 5% of that available in the experimental data. Given the actuator control signals, the NN anticipates the interactions between the unsteady flow field and aerofoil. The NN predictor has 47 inputs, 45 outputs and two hidden layers containing 32 neurons in each, which is very large. The error backpropagation (EBP) method is used until the sum squared error obtained over the training pairs decreases below a reasonably small value. Pressure values on the aerofoil are estimated by using the recordings of angle of attack and its time derivative. The NN controller has six inputs, a single output and two hidden layers containing 12 neurons in each, and the aim is a desired lift/drag response being observed. Kawthar-Ali and Acharya (1996) conducted a similar study but obtained a more marginal performance improvement. The simulation work presented in Lee *et al*. (1997) on the use of an adaptive controller based on NN to reduce drag in a turbulent channel flow predicted 20% drag reduction. Interestingly, in that study, a control scheme was derived from NN that produced the same amount of drag reduction with standard inverse control. An extended survey is presented in Kim (2003). Linear quadratic regulators, linear quadratic Gaussian controllers and adjoint-based suboptimal controls are considered. The work discusses the issues on model reduction, cost function, control laws, actuators, numerical issues and the effects of Reynolds number. In Yuen and Bau (1998), an NN-based approach to suppress chaotic convection in a thermal convection loop is presented. The NN was connected in series with the plant and it utilized the EBP algorithm to compute the weights and biases of the network. An adaptive controller developed later by the same authors has provided a better performance than this NN (Yuen and Bau, 1999). Finally, Giralt *et al*. (2000) used NNs to model the non-linear dynamics of the turbulent flow past a cylinder. The method was able to capture and identify the coherent and disordered motions in the flow.

As outlined above, some work has been done in the past decade to explore the use of NN techniques in flow modelling and control with various degrees of success. Several of these works showed promising results but were based on numerical simulations and lacked any experimental validation of the concept. The few experimental studies available are concerned with slowly varying states of the flow. Most of these works deal with models having massively interconnected structures. To the best of our knowledge, no attempt has been made so far in using an adaptive linear element (ADALINE) to model a more dynamic, higher frequency flow. Therefore, many questions remain open about the merit and effectiveness of connectionist techniques in the research on aerodynamic flows. Having this motivation in mind, in this paper, we investigated the use of an ADALINE to predict the floor pressure of a subsonic flow over a shallow rectangular cavity. A comprehensive review of this self-excited phenomenon is given by Cattafesta *et al*. (2003) and by Rowley and Williams (2006). Rossiter first developed an empirical formula for predicting the cavity flow resonance frequencies, today referred to as Rossiter frequencies or modes (Rossiter, 1964). His original model was later refined by Heller and Bliss to account for the dependence on the Mach number of the acoustic propagation (Heller and Bliss, 1975). The goal of our study is to develop an emulator predicting the floor pressure values based on localized sensory information. Clearly, the major contribution of the current paper is to demonstrate that a simple ADALINE can yield satisfactorily precise predictions with a consistent learning of the dynamics in hand. The latter is shown by considering different tuning schemes and the simplicity in structure and adaptation for a complicated experimental flow system is the original value of the presented work.

This paper is organized as follows: The second section introduces the experimental set-up. The tuning laws and the obtained simulation results are discussed in the third section, and the conclusions constitute the last part of the paper.

2. The experimental facility

In this study, the experimental facility illustrated in Figure 1 and described in detail in Debiasi and Samimy (2003, 2004) and Yan *et al*. (2006) is used. The facility consists of an optically accessible, blow-down type wind tunnel capable of continuous operation in the subsonic range. A shallow cavity with a depth $D = 12.74\,\mathrm{mm}$ and length $L = 50.8\,\mathrm{mm}$ and having length to depth aspect ratio $L/D = 4$ is recessed in the floor of the test section as shown in Figures 1–4. The cavity

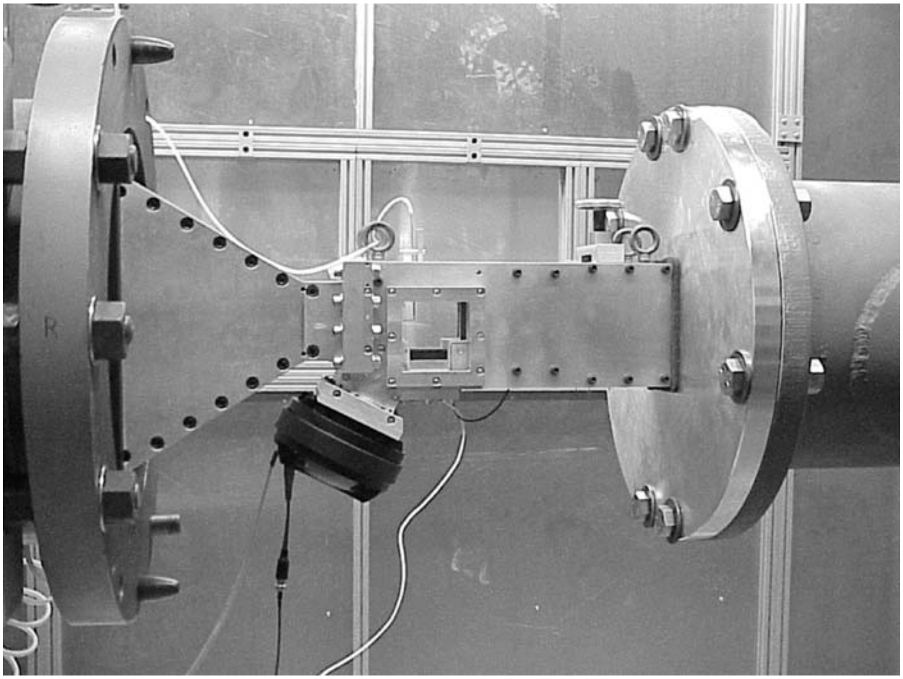

Figure 1 The appearance of the overall system

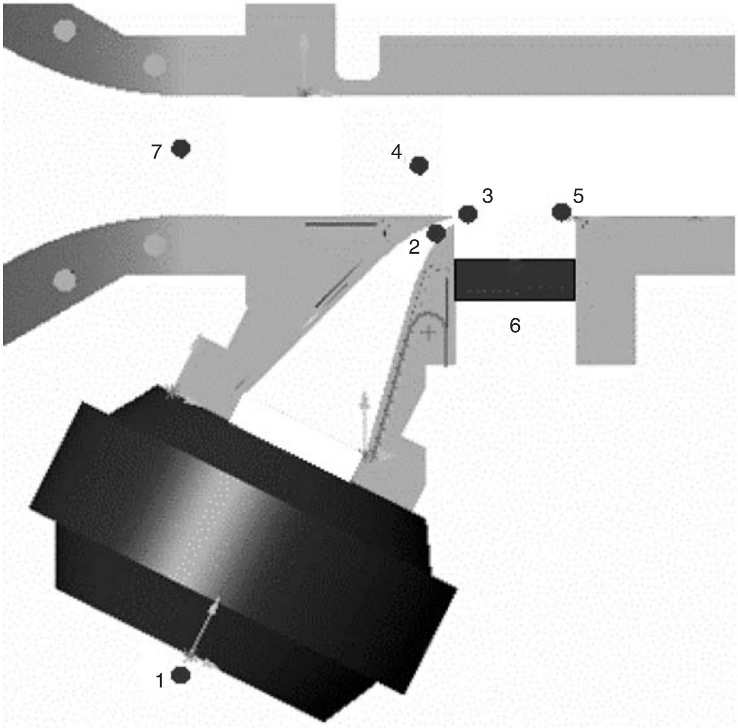

Figure 2 The physical locations of the pressure transducers

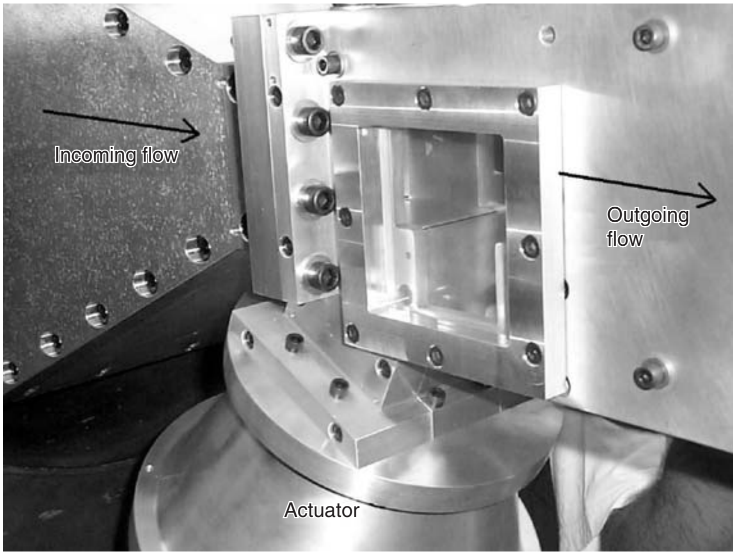

Figure 3 The incoming (baseline) flow, outgoing flow and the actuator

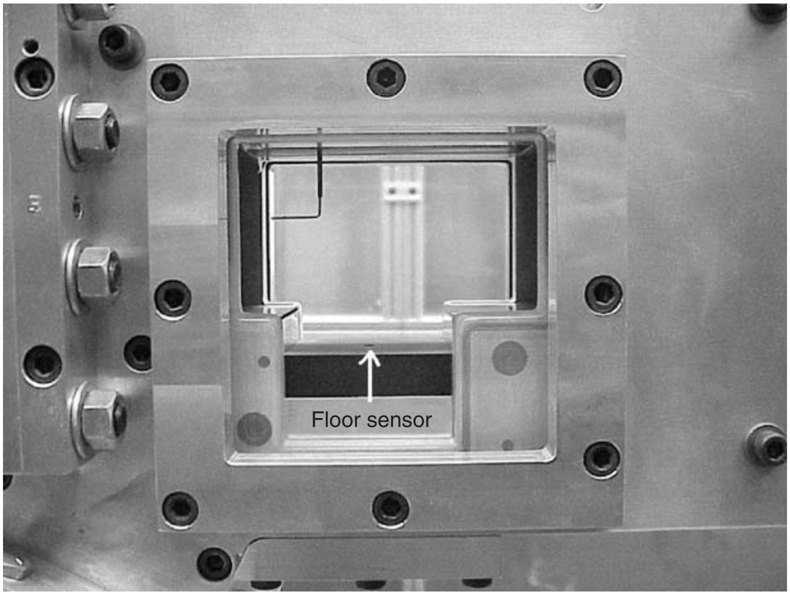

Figure 4 A close-up view of the test section with the pressure transducer placed at the centre of the cavity floor

shear layer is forced by a 2D synthetic-jet type actuator issuing from a high-aspect-ratio converging nozzle embedded in the cavity leading edge (Figures 2–4). Pressure fluctuations are measured by Kulite-type dynamic pressure transducers placed in different locations in the test section and used to derive different acoustic quantities as described in Debiasi and Samimy (2004). The control flow is provided by a synthetic jet exhausting from a slot spanning the width of the cavity upstream wall (Figure 3). The flow control set-up is equipped with a host computer, which is capable of real-time acquisition of data from the pressure transducers or hot wires, and is able to actuate a synthetic actuator shown in Figure 3 to affect the flow field in the test section. Data acquisition software with the relevant digital signal processor (DSP) board is installed on a host computer. The simultaneous time traces collected from the pressure transducers have been used to obtain the weights of the ADALINE. It is critically important to emphasize that the collected data must be spectrally rich enough to capture the cases that are likely to be encountered in real-time operation. This ensures that the tuned predictor responds appropriately to the input variables.

In Debiasi and Samimy (2004), it is observed that the cavity flow exhibits strong, single-mode resonance in the Mach number ranges 0.25–0.31 and 0.39–0.5, and multi-mode resonance in the Mach number range 0.32–0.38. It is also reported in Debiasi and Samimy (2004) that the frequency of sinusoidal forcing with the synthetic jet-like actuator has a major impact on the cavity flow resonance, whereas the effect of the amplitude is relatively minor and it affects the control authority only at higher Mach numbers.

3. Real-time conditions for the acquisition of the experimental data

As can be seen from Figure 2, simultaneous time traces can be obtained from the flow field with the aid of the DSP hardware reading the shown transducers. The sensors are located in such a way that the critical information about the flow physics is dense, ie, the entry of control excitation, the upstream and downstream wall neighbourhood, test section entry (baseline flow) and the cavity floor. In the tests, we have chosen the data read from the first, third, fifth and the sixth sensors shown in Figure 2. The first sensor records the sent excitation signal generated within the host computer. With this in mind, S_1 measures the actuation signal in volts and S_2 measures $u(t)$, the pressure fluctuations just before the actuator exit, ie, the signal equal to that produced at S_1. S_3 measures $v(t)$, the pressure fluctuation just after the actuator exit (ie, at the receptivity region at the cavity leading edge), S_4 measures the pressure fluctuations (if any) before the cavity, S_5 measures $w(t)$, the pressure fluctuations at the cavity trailing edge and S_6 measures $y(t)$, the pressure fluctuations at the centre of the cavity floor (Figure 4). Finally, S_7 measures the pressure fluctuations related to the baseline flow.

According to these definitions, we performed a set of experiments to collect the data. The first set of experiments address the noise driven cases. For Mach numbers equal to 0.25, 0.28, 0.30, 0.32 and 0.35, we have excited the flow field with a computer-generated noise signal. The excitation signal is low-pass filtered with a cut-off frequency of 200 Hz and high-pass filtered with a cut-off frequency of 10 kHz. This is implemented to meet the operating conditions of the synthetic actuator. The second set of experiments includes open-loop free forced observations at the same Mach numbers stated above. The flow field is excited with sinusoidal signals of amplitude 2.35 V_{rms} and frequency 3250 Hz and then another sinusoidal signal with 4.06 V_{rms} magnitude and frequency equal to 3920 Hz. These numbers have been set according to the open-loop expertise gained through the design of the experimental facility. Such a collective training data set clearly represents how the system behaves under different operating conditions. In all cases described above, we have adopted $T_s = 1/f_s$, where $f_s = 50$ kHz is the sampling rate, and soft-filtered the collected data with a high-pass filter having a cut-off frequency equal to 10 kHz. Among the cases described above, the case with Mach number 0.25, excitation voltage 2.35 V_{rms} and excitation frequency 3250 Hz is used only for validating the model; this dataset is not used during the model derivation.

In the next section, the modelling strategies for finding the best parameter vector $\hat{\theta}$ are discussed. Since the algorithms work in discrete time, we denote y_k for $y(t)$ measured when $t = kT_s$.

4. The ADALINE model and the adaptation schemes

The structure of the ADALINE predictor chosen in this study has the following form

$$\hat{y}_{k+1} = R_k^T \hat{\theta}_k \tag{1}$$

where

$$\hat{\theta}_k = \begin{pmatrix} a & b_1 & b_2 & c_1 & c_2 & f_1 & f_2 & g \end{pmatrix}^T \tag{2a}$$

$$R_k = \begin{pmatrix} u_k & y_k & y_{k-1} & v_k & v_{k-1} & w_k & w_{k-1} & M \end{pmatrix}^T \tag{2b}$$

where $\hat{\theta}_k$ is the parameter vector, R_k is the regressor at discrete time index k and M denotes the Mach number. The model structure given by (2a)–(2b) is determined after a set of tests seeking simplicity and performance. The goal is to find a parameter vector $\hat{\theta}$ that minimizes a cost or maximizes a performance function defined over some history of y. Since we are interested in one-step-ahead prediction, we have $d_k = y_{k+1}$, where d_k is the desired predictor output at time kT_s. So we try to minimize the error $d_k - \hat{y}_{k+1}$, in some optimal fashion, where $\hat{y}_{k+1}$ is a prediction for d_k generated from the data collected up to time kT_s.

4.1 Least mean squares algorithm (LMS)

We would like to match N successive observations of $d_k = y_{k+1}$ with the model, $\hat{y}_{k+1}$, given in (1). One can write the N observations in matrix form as

$$\begin{pmatrix} y_2 \\ \vdots \\ y_{N+1} \end{pmatrix} \approx \begin{pmatrix} R_1^T \\ \vdots \\ R_N^T \end{pmatrix} \hat{\theta} = \begin{pmatrix} \hat{y}_2 \\ \vdots \\ \hat{y}_{N+1} \end{pmatrix}, \quad Y \approx \Phi \hat{\theta} \tag{3}$$

The value of $\hat{\theta}$ minimizing $J = 1/2 \sum_{k=1}^{N} (d_k - \hat{y}_{k+1})^2$ is given by

$$\hat{\theta} = \left(\Phi^T \Phi \right)^{-1} \Phi^T Y = P_N \Phi^T Y \tag{4}$$

Notice that since the used strategy runs at batch mode, we have dropped the subscript k from $\hat{\theta}$.

4.2 Recursive least squares algorithm (RLS)

The algorithm offers an iterative solution to save the computation time and handle variations in the process likely to occur in time. Briefly, the governing equations are

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k c_k \tag{5a}$$

$$K_k = P_{k-1} R_k \left(I + R_k^T P_{k-1} R_k\right)^{-1} \tag{5b}$$

$$P_k = \left(I - K_k R_k^T\right) P_{k-1} \tag{5c}$$

where $P(t_0) = (\Phi^T \Phi)^{-1}$ is computed for some initial set of observations and the algorithm iteratively updates $\hat{\theta}_k$ based on the weighting factor K_k and the correction term $c_k := d_k - R_k^T \hat{\theta}_{k-1}$. See Aström and Wittenmark (1995) for details.

4.3 Modified Kaczmarz's algorithm (MK)

Also known as the normalized projection algorithm, the MK can also be used iteratively to achieve a good parameter vector. The update law is

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \frac{\gamma R_k}{\alpha + R_k^T R_k} \left(d_k - R_k^T \hat{\theta}_k\right) \tag{6}$$

where $\alpha \geq 0$ and $0 < \gamma < 2$. In this study, we adopt $\alpha = 1$ and $\gamma = 0.25$.

4.4 Stochastic approximation algorithm (SA)

If the data were generated by (1) with no error, the update law would be described by

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_k \left(d_k - R_k^T \hat{\theta}_k\right) \tag{7}$$

where $P_k = (\sum_{i=1}^{k} R_i^T R_i)^{-1}$. Clearly, as k increases, P_k gets smaller and smaller; consequently the update term becomes small in magnitude. Since we work on a data set with a finite number of elements, we implement the algorithm in a loop for a predefined number of executions and we record the cumulative error over all data pairs, ie, $J_i = 1/N \sum_{k=1}^{N} (d_k - R_k^T \hat{\theta}_k)^2$.

4.5 Gradient descent algorithm (GD)

Also known as the steepest descent, gradient descent is a very popular tuning algorithm in the realm of NNs. The idea that lies behind is to modify the parameter along with the negative gradient of an appropriately defined cost function, which is $J_k = 1/2e_k^2 = 1/2(d_k - \hat{y}_{k+1})^2 = 1/2(d_k - R_k^T\hat{\theta}_k)^2$. The update rule is then

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \eta \frac{\partial J_k}{\partial \hat{\theta}_k} = \hat{\theta}_k + \eta e_k R_k \tag{8}$$

where $0 < \eta < 1$ is the learning rate. To speed up the convergence, the tuning law is modified as given below

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta e_k R_k + \mu\left(\hat{\theta}_k - \hat{\theta}_{k-1}\right) \tag{9a}$$

$$\eta_{i+1} = \begin{cases} \eta_i + \Delta & E_i < E_{i-1} \\ \eta_i \zeta & E_i > E_{i-1} \end{cases} \tag{9b}$$

where (9a) describes the weight update during the *i*th forward pass (epoch) and the learning rate (η) is updated as in (9b), utilizing the value of the epoch error defined as $E_i = \sum_{k=1}^{N} J_k$. In this study, we choose $\Delta = 0$ and $\zeta = 0.99$. The algorithm typically displays very slow convergence.

4.6 Levenberg–Marquardt (LM) optimization technique

The Levenberg–Marquardt algorithm is a good balance between the GD and Newton's law. The update law is defined as below

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \left(\mu_i I + R_k R_k^T\right)^{-1} R_k e_k \tag{10a}$$

$$\mu_{i+1} = \begin{cases} \mu_i + \Delta & E_i < E_{i-1} \\ \mu_i \zeta & E_i > E_{i-1} \end{cases} \tag{10b}$$

The quantity μ_i is adapted according to the evolution of the epoch error. If μ_i is large, the above update law becomes more like the steepest descent, yet for small μ_i, the tuning behaves like Newton's law. When compared with the GD algorithm, LM predicts much better update directions at the cost of matrix inversion at each step. In this study, we choose $\Delta = 1.05$, $\zeta = 0.9$ and $\mu_0 = 1000$.

4.7 Sliding mode-based tuning algorithm (SM)

The last search strategy considered in this paper is the one introduced by Sira-Ramirez and Colina-Morles (1995). The original form of the tuning law is given by

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta \frac{R_k}{R_k^T R_k} \operatorname{sgn}\left(d_k - R_k^T \hat{\theta}_k\right) \tag{11}$$

where η is the gain of the tuning law. Because of the chattering that arises when $d_k - R_k^T\hat{\theta}_k \approx 0$, the sign function prescribes very sharp fluctuations in $\hat{\theta}_k$. For this reason, the sign function is smoothed by replacing the $\mathrm{sgn}(x) \approx x/|x| + \delta$ with $\delta > 0$ being the parameter determining the sharpness around $x = 0$. This paper uses $\delta = 0.01$ and $\eta = 0.0001$.

5. A performance comparison of the tuning laws

Several quantities are defined to distinguish the performance of the mentioned parameter search routines. The first quantity is the relative error defined by

$$e_{\mathrm{rel}} := \frac{1/N_f \sum_{k=1}^{N_f} |d_k - \hat{y}_{k+1}|^2}{1/N_f \sum_{k=1}^{N_f} |d_k|^2} \times 100\% \tag{12}$$

where $d_k = y_{k+1}$, $\hat{y}_{k+1}$ corresponds to the predicted value of the floor pressure signal, and N_f represents the "final time" at which the experiment is terminated. Note that $1/N_f \sum_{k=1}^{N_f} |d_k - \hat{y}_{k+1}|^2$ can be seen as the mean squared error (MSE) quantifying the performance over a set of observations. The second measure we are interested in is $\Delta_p := ||F\{d\} - F\{\hat{y}\}||_\infty$, where $F\{d\}$ and $F\{\hat{y}\}$ stand for the Fourier transforms of d and $\hat{y}$, respectively. The other quantities of interest are the associated computational complexity, denoted by C in Table 1, and the roots of the polynomial $D(z) = 1 - b_1 z^{-1} - b_2 z^{-2}$. Note that if we had perfect estimation, $\hat{y}_{k+1} = d_k = y_{k+1}$, then the z-transform of the autoregressive part of (1) would correspond to this polynomial and the behaviour would be described best by its roots (Equations 2a and 2b).

The tests have been carried out on the data that has not been used during the derivation of the model. The operating conditions when the test data are acquired are Mach number $M = 0.25$, the excitation signal is a sinusoidal with frequency 3250 Hz and amplitude 2.35 $\mathrm{V_{rms}}$.

According to the results summarized in the third column of Table 1, which were also presented during International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (2006), one can claim that all algorithms guide the roots of the

Table 1 Comparison metrics and obtained values

	e_{rel}	Δp	Roots of $D(z)$	MSE	C
LMS	2.786%	56.0245	$0.8843 \pm j0.2646$	8.17e–3	n/a
RLS	2.801%	53.5851	$0.8408 \pm j0.2676$	8.17e–3	High
MK	3.029%	116.9703	$0.7846 \pm j0.2073$	9.09e–3	Low
SA	2.761%	76.4459	$0.8480 \pm j0.2341$	8.22e–3	Low
GD	3.516%	245.3962	$0.7664 \pm j0.2032$	8.77e–3	Low
LM	2.838%	71.7564	$0.8216 \pm j0.2497$	8.19e–3	High
SM	2.701%	20.1625	$0.8619 \pm j0.2370$	8.25e–3	Low

polynomial $D(z)$ to values around $0.8 \pm j0.23$, which is a strong evidence of consistency of the implemented strategies. Secondly, despite the small values of the quantity e_{rel} in all strategies, the variation in Δ_{p} is significant. In terms of the peak difference (Δ_{p}), the worst case is encountered at GD and the best results are obtained with the SM algorithms. The computational intensity (C) is qualified in the last column of the above table. Since LMS algorithm is not iterative, assessing its computational needs with iterative algorithms is not of concern here. However, among the algorithms considered here, those necessitating matrix inversion introduce a greater number of computations than the others. Therefore, matrix-inverting algorithms are labelled 'high' in the last column. According to the data seen in Table 1, the smallest e_{rel} and Δ_{p} values are obtained with SM algorithm with affordable computational cost. In Figure 5, the results obtained with SM algorithm are illustrated. In the left sub-plots, the results in the frequency domain are depicted. These results indicate that the match is quite good not only around the dominant peak but also over the spectrum of interest. The right sub-plots display the results obtained in the time domain. For better illustration, we show a limited part of the entire set of results that indicate a very good prediction.

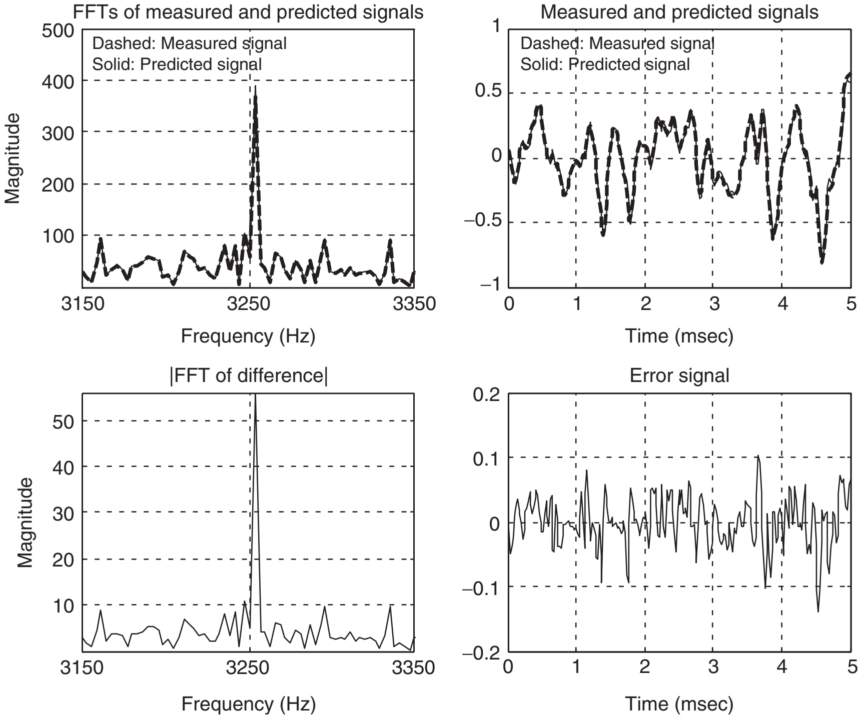

Figure 5 The results obtained with SM algorithm; 5 ms of time period results and a limited band of frequency spectrum containing the dominant peak illustrate the performance of the developed model

Aside from the quantities tabulated above, evolution of the squared norm of the parameter vector $\hat{\theta}$ is a good indicator of how the strategies update $\hat{\theta}$ and how convergent or oscillatory the parameter values are. This is illustrated in Figure 6. The results obtained emphasize that the offline implemented approaches (SA, GD, LM and SM) produce much smoother parameter norms, and among these four approaches, the earliest convergence is observed with LM. The MSE level for each algorithm is obtained with the final value of the parameter vector and are given in the fourth column of Table 1, where LMS, RLS and LM algorithms result in the smallest MSE values; however, one should note that the other values are also very close to this level.

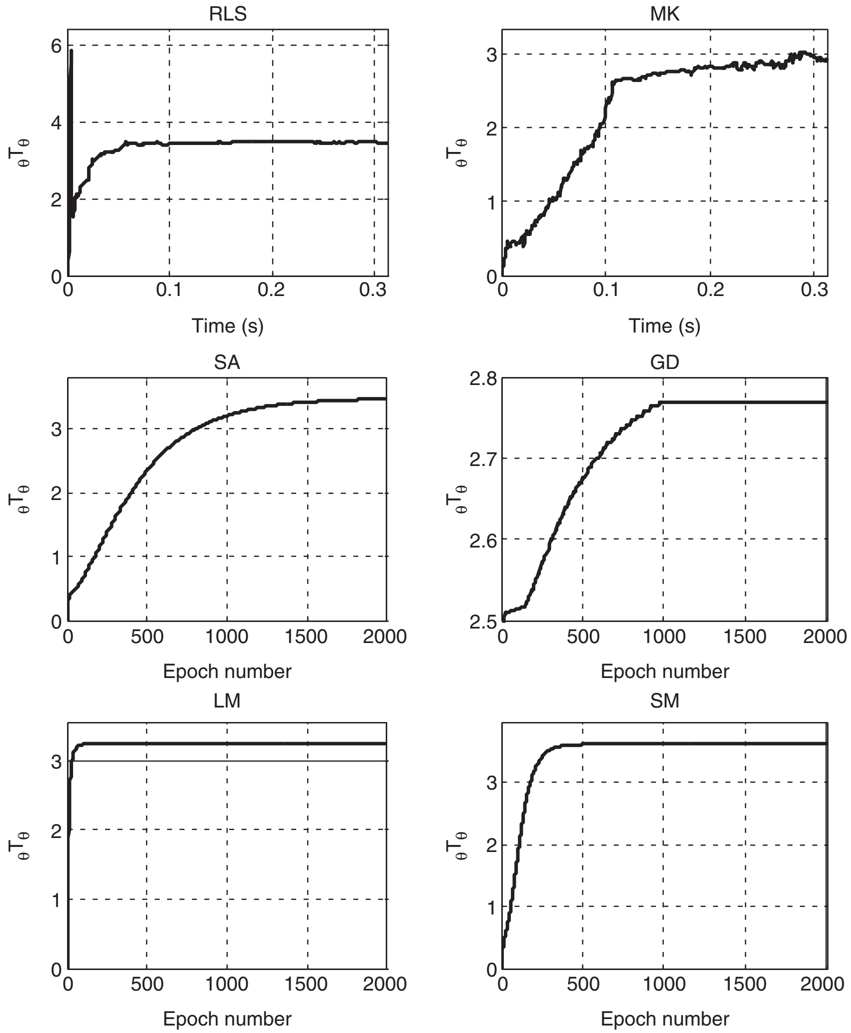

Figure 6 The norm of the parameter vector $\hat{\theta}$ as the training progresses

In terms of convergence properties, one sees that value of $\hat{\theta}^T\hat{\theta}$ settles down smoothly in the LM and SA algorithms. LM, SM and GD algorithms, on the other hand, display a visibly long flat period during which the value of $\hat{\theta}^T\hat{\theta}$ hardly changes. If the length of this period was a performance measure, then the LM and SM algorithms would be the most preferable strategies. Briefly, the evolution in all approaches except the MK is admissibly consistent.

In Figure 7, the evolution of the roots of the polynomial $D(z)$ is illustrated for each algorithm except the LMS, which runs at batch mode. At first glance, one figures out that the evolution with the MK algorithm fluctuates and the steady state values of the roots are indistinguishable. The RLS algorithm shows the next revealing fluctuations.

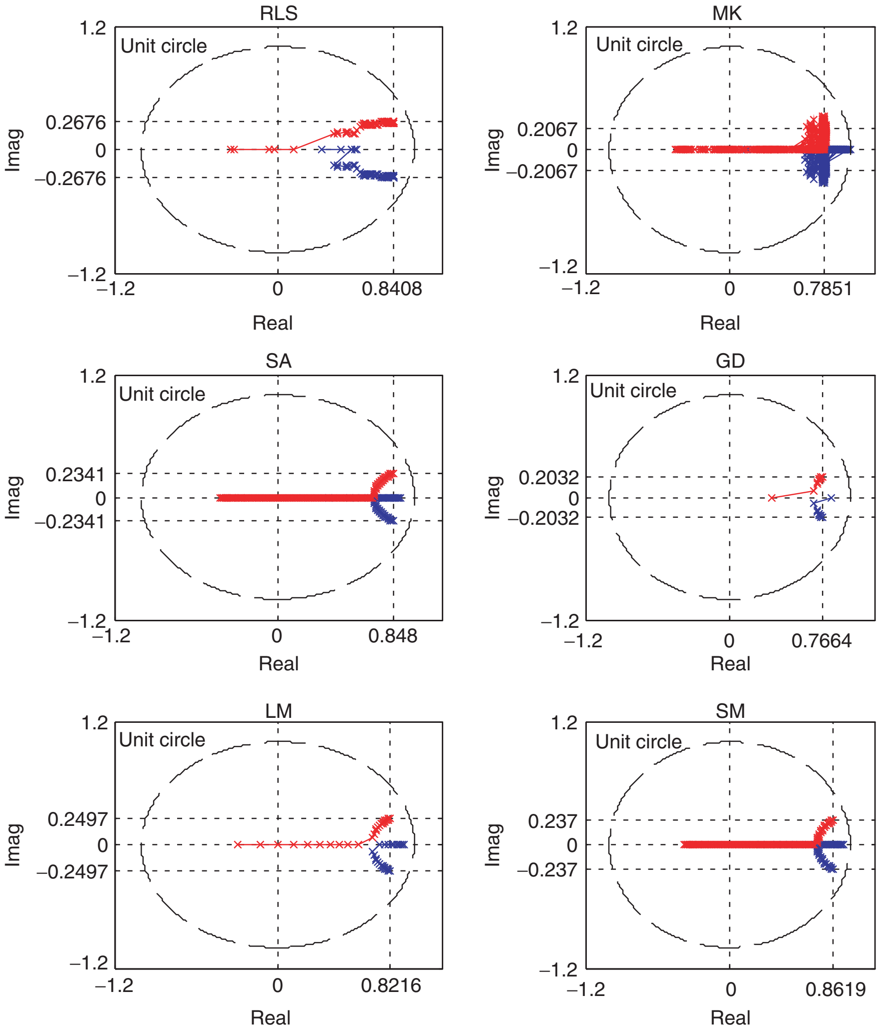

Figure 7 The roots of the polynomial $D(z)$

When the offline algorithms are considered, the GD algorithm is seen to be the one approaching the desired pole locations very quickly. Notice that the GD takes a big first update towards the goal, which is visible in Figure 7; however, it takes a long time to converge with GD and Δ_p is the largest for this scheme. In terms of big initial updates towards the goal, surprisingly the next is the LM algorithm, but its convergence rate is the best encountered in the paper (Figure 6). The remaining two strategies, namely SA and SM, come next, yet these two approaches cannot be distinguished from these subplots. Briefly, the evolution in all approaches except the MK is admissibly consistent.

A last question that should be answered here is the following: Would an algorithm be made better than the others by playing with the algorithm's parameters such as the gains η, μ, γ, α, epoch number and so on? The answer is no, based on many trials we have carried out before obtaining the best possible results, which are presented in this paper. The model could be a more complicated one possibly yielding better prediction performance; yet, the goal of this study is to demonstrate that a simple model could help in giving fairly good predictions, even revealing the capability of addressing the changes in the Mach number, which is an external input to the ADALINE.

6. Concluding remarks

This paper considers seven different algorithms for floor pressure prediction with the experimental data acquired from a shallow subsonic cavity flow facility. An ADALINE model is built based on the pressure signals read from several critical locations of the cavity. In order to see the effect of the external signals, the cavity is excited by noise and sinusoidal signals of magnitude and frequency within the allowable limits of the experimental set-up. A relative error is defined to compare the ratio of the powers of reconstruction error signal and the measured pressure signal. Similarly, the peak value of the FFT magnitude of the reconstruction error is also checked for all methods. These two quantities are effective in deciding the best approach, which points to the SM algorithm. The computational cost of the SM algorithm is low and the roots of $D(z)$ follow a very smooth path in the complex plane. In conclusion, although we make no claim that the SM algorithm is superior to the others in all such applications, the SM algorithm's desirable features are visible from the obtained results and the structure of the predictor is simple enough to consider in real-time applications requiring pressure predictions based on localized information.

References

Agarwal, M. 1997: Combining neural and conventional paradigms for modelling prediction and control. *International Journal of Systems Science* 28, 65–81.

Aström, K.J. and **Wittenmark, B.** 1995: *Adaptive control*. Addison Wesley.

Cattafesta, L., Williams, D., Rowley, C.W. and **Alvi, F.** 2003: Review of active control of flow-induced cavity resonance. *33rd AIAA Fluid Dynamics Conference*, Orlando FL, 23–26 June, AIAA 2003–3567.

Debiasi, **M**. and **Samimy**, **M**. 2003: *An experimental study of the cavity flow for closed-loop flow control*. AIAA Paper 2003-4003.

Debiasi, M. and **Samimy, M.** 2004: Logic-based active control of subsonic cavity flow resonance. *AIAA Journal* 42, 1901–909.

Efe, **M.Ö.**, **Debiasi**, **M.**, **Yan**, **P.**, **Özbay**, **H**. and **Samimy**, **M**. 2005a: Control of subsonic cavity flows by neural networks – analytical models and experimental validation. *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 10–13 January, Reno, NV, AIAA 2005–0294.

Efe, **M.Ö.**, **Debiasi**, **M.**, **Yan**, **P.**, **Özbay**, **H**. and **Samimy**, **M**. 2005b: A generalizing fuzzy model for shallow cavity flows under different mach regimes. *2005 IEEE Conference on Control Applications (CCA'2005)*, 28–31 August, Toronto, 67–72.

Faller, **W.E.**, **Schreck**, **S.J**. and **Luttges**, **M.W**. 1994: Real-time prediction and control of three dimensional unsteady separated flow fields using neural networks. *32nd Aerospace Sciences Meeting and Exhibit*, 10–13 January, Reno, NV, AIAA 94–0532.

Fan, **X.**, **Hofmann**, **L**. and **Herbert**, **T**. 1993: Active flow control with neural networks. *AIAA Shear Flow Conference*, 6–9 July, Orlando, FL, AIAA 93–3273.

Giralt, F., **Arenas, A.**, **Ferre-Gine, J.**, **Rallo, R.** and **Kopp, G.A.** 2000: the simulation and interpretation of free turbulence with a cognitive neural system. *Physics of Fluids 12*, 1826–35.

Heller, **H.H**. and **Bliss**, **D.B**. 1975: *The physical mechanisms of flow-induced pressure fluctuations in cavities and concepts for their suppression*. AIAA Paper 75–491.

Jacobson, **S.A**. and **Reynolds**, **W.C**. 1993: Active control of boundary layer wall shear stress using self-learning neural networks. *AIAA Shear Flow Conference*, Orlando, FL, 6–9 July, AIAA 93–3272.

Kawthar-Ali, **M.H.** and **Acharya**, **M**. 1996: Artificial neural networks for suppression of the dynamic stall vortex over pitching airfoils. *34th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 15–18 January, AIAA 96–0540.

Kim, J. 2003: Control of turbulent boundary layers. *Physics of Fluids* 15, 1093–105.

Lee, C., **Kim, J.**, **Babcock, D.** and **Goodman, R.** 1997: Application of neural networks to turbulence control for drag reduction. *Physics of Fluids* 9, 1740–47.

Narendra, K.S. and **Parthasarathy, K.** 1990: Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks* 1, 4–27.

Rowley, **C.W.**, **Colonius**, **T**. and **Murray**, **R.M**. 2001: Dynamical models for control of cavity oscillations. *7th AIAA/CEAS Aeroacoustics Conference*, Maastricht, 28–30 May, AIAA 2001–2126.

Rowley, C.W. and **Williams, D.R.** 2006: Dynamics and control of high-Reynolds-number flow over open cavities. *Annual Review of Fluid Mechanics* 38, 251–76.

Rossiter, **J.E**. 1964: *Wind tunnel experiments on the flow over rectangular cavities at subsonic and transonic speeds*. RAE Tech Rep 64037 1964 and Aeronautical Research Council Reports and Memoranda No 3438.

Sira-Ramirez, H. and **Colina-Morles, E.** 1995: A sliding mode strategy for adaptive learning in ADALINES. *IEEE*. *Transactions on Circuits and Systems – I: Fundamental Theory and Applications* 42, 1001–12.

Yan, P., **Debiasi, M.**, **Yuan, X.**, **Little, J.**, **Özbay, H.** and **Samimy, M.** 2006: Experimental study of linear closed-loop control of subsonic cavity flow. *AIAA Journal 44*, 929–38.

Yuan, X., **Efe, M.Ö.** and **Özbay, H.** 2004: On delay-based linear models and robust control of cavity flows. In Niculescu, S. and Gu, K., editors. *Advances in time-delay systems*. Springer, LNCSE, 38, 287–98.

Yuan, X., **Caraballo, E.**, **Yan, P.**, **Özbay**, **H.**, **Serrani, A.**, **DeBonis**, **J.**, **Myatt**, **J.H**. and **Samimy**, **M**. 2005: Reduced order model based feedback controller design for subsonic cavity flows. *AIAA Aerospace Science* Meeting, Reno, NV, 10–13 January.

Yuen, P.K. and **Bau, H.H.** 1998: Controlling chaotic convection using neural nets – theory and experiments. *Neural Networks 11*, 557–69.

Yuen, P.K. and **Bau, H.H.** 1999: Optimal and adaptive control of chaotic convection – theory and experiments. *Physics of Fluids 11*, 1435–48.