Iterative and Batch Mode Algorithms for Modeling of Aerodynamic Cavity Flows Under Boundary Excitations*

M. Önder Efe[1] Marco Debiasi[2] Peng Yan[3] Hitay Özbay[4] Mohammad Samimy[5]

Abstract— **Modeling of cavity flows is an interesting research field as the system under investigation is nonlinear and spatially distributed. This paper focuses on modeling based on pointwise observations under different boundary excitations controlling the flow oscillations. Various 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 Algorithm (LM) and Sliding Mode Based Tuning (SM). A simple model and its properties are discussed comparatively.**

I. Introduction

Modeling efforts for aerodynamic flows have primarily the goal of obtaining a good model which is to be used for feedback control system design. The results presented in [1]-[4] emphasize the fact that there are several alternatives towards the goal, yet the suitability of each approach depends on the operating condition and the flow geometry in hand. One approach for modeling is to exploit the physics of the problem, [1]-[3]. Individual submodels are developed 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. Another major research direction focuses on Proper Orthogonal Decomposition (POD) for modeling of shallow cavity flows, [4]-[5]. A detailed review of the flow induced cavity oscillations is presented in [6].

This paper describes a third alternative for modeling 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 on 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 modeling and control goals become achievable. In our earlier studies neural networks, [7], and fuzzy logic, [8], are shown to be useful techniques in flow modeling. Whereas the approaches considered in this paper yield a fine tuned representative dynamic model which has a significantly simpler structure than those obtained through the use of neural networks and fuzzy logic.

This paper is organized as follows. The second section introduces the experimental setup. The third section describes how the experimental data is acquired. The fourth section presents the utilized modeling techniques, the fifth section discusses the comparison of the modeling strategies, and the conclusions constitute the last part of the paper.

II. The Experimental Facility

In this study, the experimental facility illustrated in Fig. 1 and described in detail in [9]-[10] 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 $D = 12.74$ mm and length $L = 50.8$ mm and having length to depth aspect ratio $L/D = 4$ is recessed in the floor of the test section. The cavity shear-layer is forced by a 2-D synthetic-jet type actuator issuing from a high-aspect-ratio converging nozzle embedded in the cavity leading edge. 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 [10]. The control flow is provided by a synthetic jet exhausting from a slot spanning the width of the cavity upstream wall (See Fig. 1 and 2). The flow control setup is equipped with a 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 to affect the flow field in the test section. A data acquisition software with relevant Digital Signal Processor (DSP) board is installed on the host computer running the controller that closes the loop. The simultaneous time traces collected from the pressure transducers have been used to obtain the representative model. 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 makes it sure that the devised model responds appropriately to the input variables.

*This work was supported in part by AFRL/VA and AFOSR under contract no F33615-01-2-3154 and in part by the European Commission under contract no. MIRG-CT-2004-006666 and in part by TOBB University of Economics and Technology, BAP Program.

[1] Corresponding Author, TOBB University of Economics and Technology, Department of Electrical and Electronics Engineering, Sögütözü, Ankara, Turkey, E-mail: onderefe@ieee.org, Phone: +90-312-292-4064, Fax: +90-312-292 4091

[2] Department of Mechanical Engineering, The Ohio State University, Columbus, OH 43210, U.S.A., E-mail: debiasi.1@osu.edu

[3] was with Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA; present address: Seagate Tech Center, Bloomington MN 55435 USA, E-mail: yan.39@osu.edu, Peng.Yan@seagate.com

[4] Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800, Ankara, Turkey, on leave from Department of Electrical and Computer Eng., The Ohio State University, E-mail: hitay@bilkent.edu.tr, ozbay@ece.osu.edu

[5] Department of Mechanical Engineering, The Ohio State University, Columbus, OH 43210, U.S.A., E-mail: samimy.1@osu.edu

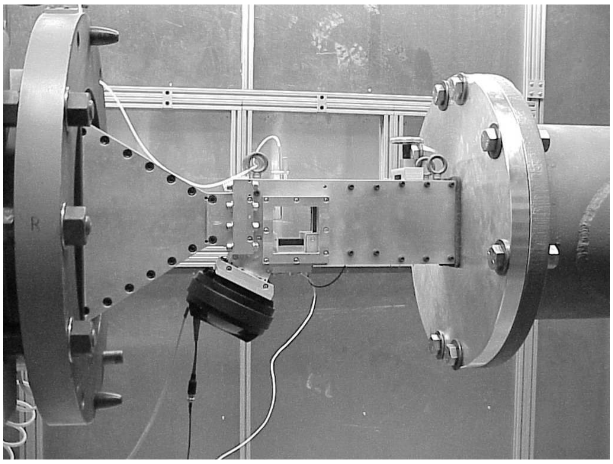

Fig. 1. The appearance of the overall system

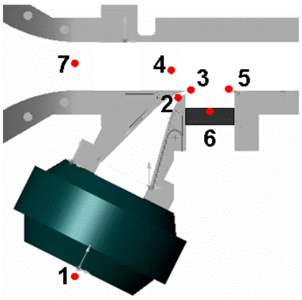

Fig. 2. The locations of the transducers placed in the test section

In [10], 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 [10] 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.

III. Acquisition of the Experimental Data

As can be seen from Fig. 2, simultaneous time traces can be obtained from the flow field by the aid of the DSP hardware. The sensors are located in such a way that the critical information about the flow physics is dense, i.e. the entry of control excitation, the upstream and downstream wall neighborhood, 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 Fig. 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, S_2 measures $u(t)$, the pressure fluctuations just before the actuator exit, i.e. the signal equal to that produced at S_1. S_3 measures $v(t)$, the pressure fluctuation just after the actuator exit (i.e. 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, S_6 measures $y(t)$, the pressure fluctuations at the center of the cavity floor. Finally, S_7 measures the pressure fluctuations due 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 computer generated noise signal. The excitation signal is lowpass filtered with cutoff frequency 200 Hz and highpass filtered with cutoff frequency 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 $2.35V_{\text{rms}}$ and frequency 3250 Hz and then another sinusoidal signal with $4.06V_{\text{rms}}$ 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 highpass filter having cutoff frequency equal to 10 kHz. Among the cases described above, the case with Mach number equals 0.25, excitation voltage is $2.35V_{\text{rms}}$ and excitation frequency is 3250 is used only for validating the model, and this data is not used during the model derivation.

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

IV. Modeling Strategies

The model chosen in this study has the following form

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

where

$$\hat{\theta}_k = (a \quad b_1 \quad b_2 \quad c_1 \quad c_2 \quad f_1 \quad f_2 \quad g)^{\mathrm{T}} \tag{2a}$$

$$R_k = (u_k \quad y_k \quad y_{k-1} \quad v_k \quad v_{k-1} \quad w_k \quad w_{k-1} \quad \mathrm{M})^{\mathrm{T}} \tag{2b}$$

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 goal is to find a parameter vector $\hat{\theta}$ that minimizes some cost or maximizes some performance function defined over some history of y. We define d to be the desired predicted value of y. Typically, we are interested in one-step-ahead prediction, i.e., $d_k = y_{k+1}$. So we try to minimize the error $d_k - \hat{y}_{k+1}$, in some optimal fashion, where $\hat{y}_{k+1}$ is the estimation of d_k generated from data collected up to time $t = kT_s$ via (1).

A. 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

$$\begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_{N+1} \end{pmatrix} \approx \begin{pmatrix} R_1^{\mathrm{T}} \\ R_2^{\mathrm{T}} \\ \vdots \\ R_N^{\mathrm{T}} \end{pmatrix} \hat{\theta} = \begin{pmatrix} \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_{N+1} \end{pmatrix} \tag{3a}$$

$$\mathbb{Y} \approx \Phi\hat{\theta} \tag{3b}$$

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

$$\begin{aligned} \hat{\theta} &= \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\mathbb{Y} \\ &= \mathrm{P}_N\Phi^{\mathrm{T}}\mathbb{Y}. \end{aligned} \tag{4}$$

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

B. Recursive Least Squares Algorithm (RLS)

Since in real time the data are read continuously, Recursive Least Squares (RLS) 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

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k\left(d_k - R_k^{\mathrm{T}}\hat{\theta}_{k-1}\right) \tag{5a}$$

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

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

where $P_{t_0} = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}$ 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 $d_k - R_k^{\mathrm{T}}\hat{\theta}_{k-1}$ (See [11] for details).

C. Modified Kaczmarz's Algorithm (MK)

Also known as the Normalized Projection Algorithm, the modified Kaczmarz's algorithm can also be used iteratively to achieve a good parameter vector. The update law is

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

where $\alpha \geq 0$ and $0 < \gamma < 2$. In this study, we adopt $\alpha = 1$ and $\gamma = 0.25$. A pseudo code for the implementation of the algorithm is presented at the end of the section.

D. Stochastic Approximation Algorithm (SA)

The algorithm assumes that the data are generated by (1) with no error, and the update law is described by

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

where $P_k = \left(\sum_{i=1}^{k} R_i^{\mathrm{T}}R_i\right)^{-1}$. Clearly, as k increases, P_k gets smaller and smaller, consequently the update term gets small in magnitude. Since we work on a data set with 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, i.e. $J_i = \frac{1}{N}\sum_{k=1}^{N}\left(d_k - R_k^{\mathrm{T}}\hat{\theta}_k\right)^2$.

E. Gradient Descent Algorithm (GD)

Also known as the steepest descent, gradient descent is a very popular tuning algorithm in the realm of neural networks. The idea that lies behind is to modify the parameter along with the negative gradient of an appropriately defined cost function, which is

$$\begin{aligned} J_k &= \frac{1}{2}e_k^2 = \frac{1}{2}\left(d_k - \hat{y}_{k+1}\right)^2 \\ &= \frac{1}{2}\left(d_k - R_k^{\mathrm{T}}\hat{\theta}_k\right)^2 \end{aligned}$$

The update rule is then

$$\begin{aligned} \hat{\theta}_{k+1} &= \hat{\theta}_k - \eta\frac{\partial J_k}{\partial\hat{\theta}_k} \\ &= \hat{\theta}_k + \eta e_kR_k, \end{aligned} \tag{8}$$

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

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

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

where (9a) is for the i-th forward pass (epoch), and the learning rate is updated according to 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, [12].

F. Levenberg-Marquardt Algorithm (LM)

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

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \left(\mu_iI + R_kR_k^{\mathrm{T}}\right)^{-1}R_ke_k \tag{10a}$$

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

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 Netwon's law. When compared with 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 $\mu_0 = 1000$.

G. Sliding Mode Based Tuning Algorithm (SM)

The last search strategy considered in this paper is the one introduced by Sira-Ramirez et al, [13]. The original form of the tuning law is given by

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

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

```
% A pseudo code for the implemented algorithms
for $i$ = 1 to 2000
    Set $E_i \leftarrow 0$
    for $k$ = 1 to $N$
        $e_k = d_k - R_k^{\mathrm{T}} \hat{\theta}_k$
        Execute (6) for MK
        Execute (7) for SA
        Execute (9a) for GD
        Execute (10a) for LM
        Execute (11) for SM
        Set $E_i \leftarrow E_i + e_k^2$
    end
    Execute (9b) for GD, (10b) for LM
end
```

V. Comparison of the Search Strategies

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

$$e_{\mathrm{rel}} := \frac{\frac{1}{N_f} \sum_1^{N_f} |d_k - \hat{y}_{k+1}|^2}{\frac{1}{N_f} \sum_1^{N_f} |d_k|^2} \times 100 \quad \% \qquad (12)$$

where $d_k = y_{k+1}$ and $\hat{y}_{k+1}$ 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 $\frac{1}{N_f} \sum_1^{N_f} |d_k - \hat{y}_{k+1}|^2$ can be seen as the mean square error (MSE). The second measure we are interested in is

$$\Delta_p := \|\mathcal{F}\{d\} - \mathcal{F}\{\hat{y}\}\|_\infty$$

where $\mathcal{F}\{d\}$ stands for the Fourier transform of d, and similarly for $\mathcal{F}\{\hat{y}\}$. The other quantities of interest are the computational complexity, denoted by $\mathcal{C}$ in the table below, and the roots of the polynomial $\mathcal{D}(z) = 1 - b_1 z^{-1} - b_2 z^{-2}$ (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).

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 and the excitation signal is a sinusoidal with frequency 3250 Hz and amplitude $2.35 V_{\mathrm{rms}}$.

TABLE I
Comparison Metrics and Obtained Values

	e_{rel}	Δ_p	Roots of $\mathcal{D}(z)$	MSE	$\mathcal{C}$
LMS	2.786%	56.0245	$0.8843 \pm j0.2646$	n/a	n/a
RLS	2.801%	53.5851	$0.8408 \pm j0.2676$	n/a	High
MK	3.029%	116.9703	$0.7846 \pm j0.2073$	n/a	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%	26.3429	$0.8619 \pm j0.2370$	8.25e-3	Low

According to the results summarized in the third column of Table I, one can claim that all algorithms guide the roots of the polynomial $\mathcal{D}(z)$ 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, the worst case is encountered at GD and the best case is obtained with the SM algorithms. The computational intensity ($\mathcal{C}$) 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 concerned here. However, among the algorithms considered here, those necessitating matrix inversion introduce more number of computations than the others. Therefore, matrix inverting algorithms are labeled High in the last column. According to the data seen in Table I, the smallest e_{rel} and Δ_p values are obtained with SM algorithm with affordable computational cost. In Fig. 3, the results obtained with SM algorithm are illustrated. In the left subplots, the results in 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 subplots display the results obtained over a very short time period, indicating a good reconstruction.

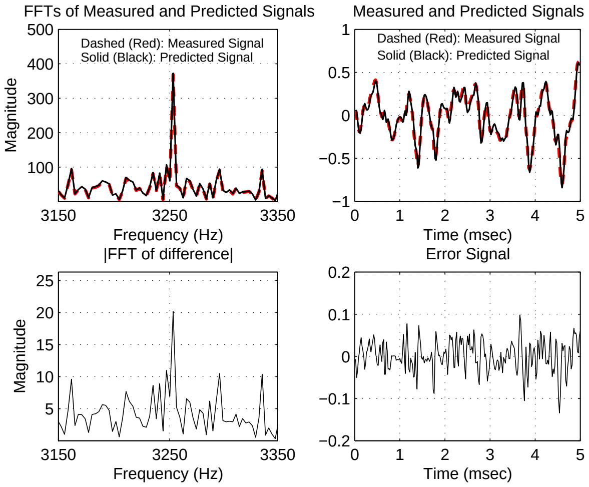

Fig. 3. The results obtained with SM algorithm. 5 milliseconds of time period results and a limited band of frequency spectrum containing the dominant peak illustrate the performance of the developed model.

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. In Fig. 4, we illustrate the quantity $\hat{\theta}^{\mathrm{T}}\hat{\theta}$ for each approach. Since RLS and MK are online approaches, the horizontal axis in these subplots are time in milliseconds. The algorithms SA, GD, LM and SM are implemented offline and the horizontal axes of these plots are epochs, i.e. the number of forward and backward passes of all training samples once. These algorithms are executed for 2000 epochs, and after this number the parameter update is stopped.

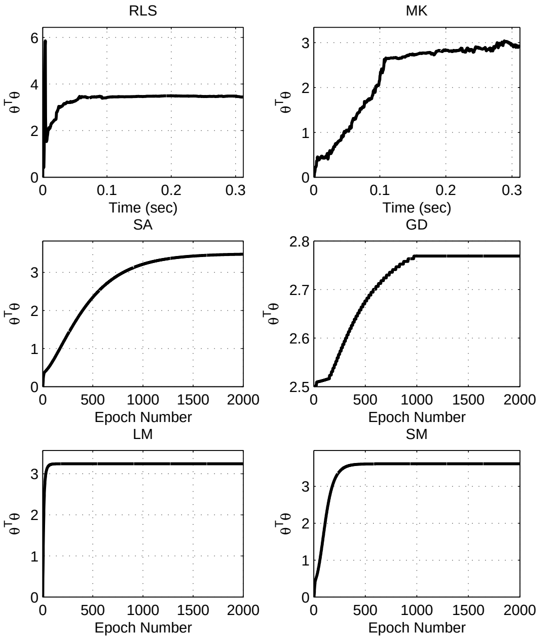

Fig. 4. The evolution of $\hat{\theta}^{\mathrm{T}}\hat{\theta}$ over discrete time frames (seconds or epochs)

The results shown in Fig. 4 emphasize that the offline implemented approaches (SA, GD, LM and SM) produce much smoother parameter norms and among these four approaches, earliest convergence is observed with LM. MSE levels for these algorithms are also given in the fourth column of Table I, where LM algorithm results in the smallest MSE value, however, one should note that the other values are also very close to this value. In terms of convergence properties, one sees that value of $\hat{\theta}^{\mathrm{T}}\hat{\theta}$ 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 $\hat{\theta}^{\mathrm{T}}\hat{\theta}$ hardly changes. If the length of this period was a performance measure, than LM and SM algorithms would be the most preferable strategies.

In Fig. 5, the evolution of the roots of the polynomial $\mathcal{D}(z)$ is illustrated for each algorithm except the LMS which runs at batch mode. At a first glance, one figures out that the evolution with MK algorithm fluctuating and the steady state values of the poles are indistinguishable. The RLS algorithm is the next revealing fluctuations. When the offline algorithms are considered, the GD algorithm is seen to be the one approaching the desired pole locations very quickly.

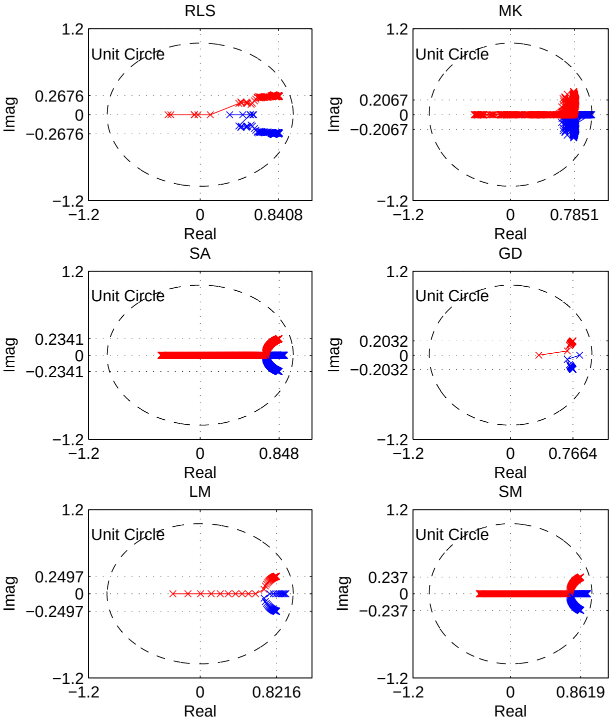

Fig. 5. Locus of the roots of $\mathcal{D}(z)$ in the complex plane

Notice that the GD takes a big first update towards the goal, which is visible in Fig. 5, however, it takes a long time to converge with GD and Δ_p is the largest seen 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 (See Fig. 4). The remaining two strategies, namely SA and SM comes next yet these two approaches cannot be distinguished from these subplots. Briefly, the evolution in all approaches except the MK is admissibly consistent.

TABLE II

Final Values of Parameters

Parameter	Value
a	0.0245
b_1	1.7238
b_2	-0.7990
c_1	-0.0234
c_2	1.0843e-3
f_1	4.8068e-3
f_2	-38.5801e-3
g	-4.6581e-3

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 done before obtaining the best possible results, which are presented in this paper.

VI. Conclusions

This paper considers seven different algorithms for modeling of aerodynamic cavity flows. A dynamical 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 setup. 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 the SM algorithm. The computational cost of SM algorithm is low and the roots of the autoregressive part polynomial follow a very smooth path in the complex plane. In conclusion, the SM algorithm's desirable features are visible from obtained results, however, it is difficult to claim that SM is superior to the others in all applications.

References

[1] Williams, D.R., Rowley C.W., Colonius, T., Murray R.M., MacMartin, D.G., Fabris, D., Albertson, J., "Model Based Control of Cavity Oscillations Part I: Experiments," 40th Aerospace Sciences Meeting (AIAA 2002-0971), Reno, NV, 2002.

[2] Rowley, C.W., Williams, D.R., Colonius, T., Murray, R.M., MacMartin, D.G., Fabris, D., "Model Based Control of Cavity Oscillations Part II: System Identification and Analysis," 40th Aerospace Sciences Meeting (AIAA 2002-0972), Reno, NV, 2002.

[3] Rowley, C.W., Colonius, T., Murray, R.M., "Dynamical Models for Control of Cavity Oscillations," 7th AIAA/CEAS Aeroacoustics Conf. (AIAA 2001-2126), May 28-30, Maastricht, The Netherlands, 2001.

[4] Rowley, C.W., Colonius, T., Murray, R.M., "Model Reduction for Compressible Flows, Using POD and Galerkin Projection" *Physica D*, v.189, pp.115-129, 2004.

[5] Yuan, X., Caraballo, E., Yan, P., Özbay, H., Serrani, A., DeBonis, J., Myatt, J.H., Samimy, M., "Reduced Order Model Based Feedback Controller Design for subsonic Cavity Flows," AIAA Aerospace Science Meeting, Jan. 10-13, Reno, NV, 2005.

[6] Cattafesta, L., Williams, D., Rowley, C.W., Alvi, F., "Review of Active Control of Flow-Induced Cavity Resonance," 33rd AIAA Fluid Dynamics Conference, June 23-26, 2003 Orlando, FL, AIAA2003-3567, 2003.

[7] Efe, M.Ö., Debiasi, M., Yan, P., Özbay, H., Samimy, M., "Control of Subsonic Cavity Flows by Neural Networks - Analytical Models and Experimental Validation," 43rd AIAA Aerospace Sciences Meeting and Exhibit, January 10-13, 2005, Reno, NV, AIAA2005-0294, 2005.

[8] Efe, M.Ö., Debiasi, M., Yan, P., Özbay, H., Samimy, M., "A Generalizing Fuzzy Model for Shallow Cavity Flows Under Different Mach Regimes," 2005 IEEE Conf. on Control Applications (CCA'2005), August 28-31, Toronto, Canada, pp.67-72, 2005.

[9] Debiasi, M., and Samimy, M., "An Experimental Study of the Cavity Flow for Closed-Loop Flow Control," AIAA Paper No: 2003-4003, June 2003.

[10] Debiasi, M. and Samimy, M., "Logic-Based Active Control of Subsonic Cavity Flow Resonance," *AIAA Journal*, Vol. 42, No. 9, pp. 1901-1909, September 2004.

[11] Åström, K.J., Wittenmark, B., *Adaptive Control*, Addison Wesley, 1995.

[12] Haykin, S., *Neural Networks*, Macmillan College Printing Company, New Jersey, 1994.

[13] Sira-Ramirez, H. and E. Colina-Morles, "A Sliding Mode Strategy for Adaptive Learning in Adalines," *IEEE Trans. on Circuits and Systems - I: Fundamental Theory and Applications*, v.42, no.12, pp.1001-1012, 1995.